UNIVERSITY OF GENOVA

PHD PROGRAM IN COMPUTER VISION, PATTERN RECOGNITION AND MACHINE LEARNING

Coping with Data Scarcity in Deep Learning and Applications for Social Good

by

Matteo Bustreo

Thesis submitted for the degree of *Doctor of Philosophy* (XXXIII cycle)

November 2020

Supervisor Vittorio Murino

Ai miei genitori

Acknowledgements

Non capita spesso l'opportunità di fermarsi per ringraziare chi ha condiviso con noi un lungo percorso e ci ha dedicato tempo, consigli ed incoraggiamenti.

Quale migliore occasione della Tesi di Dottorato?

Innanzitutto ringrazio il mio Supervisore, **Prof. Vittorio Murino**, che non soltanto mi ha messo a disposizione la sua esperienza e competenza in questi anni, ma soprattutto mi ha dato la possibilità di intraprendere questo impegnativo percorso.Ringrazio il nuovo Principal Investigator di PAVIS, **Alessio Del Bue**, che mi ha permesso di continuare con l'attività di ricerca impostata prima del suo arrivo e contribuito con i suoi consigli alla buona riuscita di molte delle attività presentate in questo lavoro. Ringrazio i moltissimi colleghi/amici che sono stati al mio fianco in questi 10 anni in IIT, dandomi prima motivazione e poi aiuto e supporto durante il Dottorato. A partire dalla vecchia guardia: **Pietro Chaltron**, **Michele**, **Raghu**, **Reza**, **Matteo**, **Marco Sanbi**, **Minh**, **Croccolus**, **Seba**, **Samu**, **Loris**, **Luca**, **Silvia**, **Gianca Enver**, **Cosimo**, **Simona**, **Paolo**, **Behzad**; arrivando ai nuovi ricercatori, che forse mi considerano una persona seria: **Avik**, **Maya**, **Veronica**, **Noman**, **Cigdem**, **Nicolò**, **Jacopo**, **Milind**, **Diego**, **Pietro**, **Yiming**, **Gian Luca**, **Shahid**, **Andrea**, **Riccardo**, **Nuno**. Un ringraziamento particolare al mio sfogatoio e compagno di banco ormai da 8 anni, **Carlos**. Sono stati 10 anni impegnativi ma divertenti. Ho avuto la possibilità di confrontarmi con persone provenienti da tutto il mondo che, dopo un breve periodo di condivisione, sono tornate a sparpagliarsi, ognuno per la propria strada. E proprio grazie alla tecnologia riusciamo ancora a rimanere in contatto e scambiarci consigli e prese in giro.

Se sono in IIT è anche grazie alle moltissime cose che ho imparato in ufficio e in trasferta dai miei vecchi colleghi della **Microtec** e, ancor prima, dall'opportunità che mi è stata data alla UCLA dai miei relatori di Tesi Magistrale, **Prof.ssa Maria Elena Valcher** e **Prof. Stefano Soatto**.

Ma non penso che sarei stato altrettanto felice di seguire il percorso che ho fatto se non avessi avuto la serenità che deriva dalla fortuna di essere circondato da un gruppo di amici che si sono sempre fatti sentire vicini, indipendentemente dalle distanze fisiche: il **Gargamella's**

Fan Club, il **Gruppo Vacanze INPS**, gli **amici storici dalla prole inarrestabile**, la **Genova Ultimate Disc**.

Il ringraziamento più grosso va ovviamente alle persone che sono per me le più importanti e che mi conoscono meglio di quanto mi conosca io stesso. I miei genitori **Lucio** e **Caterina**, che mi hanno spronato fin da quando passavo i pomeriggi a fare i "compiti per casa"; mia sorella **Cecila**, esempio quotidiano, fonte inesauribile di incentivi con le sue prese per il *cool* e ladra di amici; mia moglie **Laura**, stimolo e confronto giornaliero, inesauribile fonte di iniziative e avventure, senza il cui appoggio e sostegno non avrei potuto raggiungere questo traguardo e probabilmente nemmeno iniziarlo; **Elide**, che trasforma in divertimento e gioco anche i periodi più impegnativi e stressanti; infine **...**, che per il momento è solo . ma è già molto importante.

Grazie a tutti!

Abstract

The recent years are experiencing an extremely fast evolution of the Computer Vision and Machine Learning fields: several application domains benefit from the newly developed technologies and industries are investing a growing amount of money in Artificial Intelligence. Convolutional Neural Networks and Deep Learning substantially contributed to the rise and the diffusion of AI-based solutions, creating the potential for many disruptive new businesses.

The effectiveness of Deep Learning models is grounded by the availability of a huge amount of training data. Unfortunately, data collection and labeling is an extremely expensive task in terms of both time and costs; moreover, it frequently requires the collaboration of domain experts.

In the first part of the thesis, I will investigate some methods for reducing the cost of data acquisition for Deep Learning applications in the relatively constrained industrial scenarios related to visual inspection. I will primarily assess the effectiveness of Deep Neural Networks in comparison with several classical Machine Learning algorithms requiring a smaller amount of data to be trained. Hereafter, I will introduce a hardware-based data augmentation approach, which leads to a considerable performance boost taking advantage of a novel illumination setup designed for this purpose. Finally, I will investigate the situation in which acquiring a sufficient number of training samples is not possible, in particular the most extreme situation: zero-shot learning (ZSL), which is the problem of multi-class classification when no training data is available for some of the classes. Visual features designed for image classification and trained offline have been shown to be useful for ZSL to generalize towards classes not seen during training. Nevertheless, I will show that recognition performances on unseen classes can be sharply improved by learning ad hoc semantic embedding (the pre-defined list of present and absent attributes that represent a class) and visual features, to increase the correlation between the two geometrical spaces and ease the metric learning process for ZSL.

In the second part of the thesis, I will present some successful applications of state-of-the-art Computer Vision, Data Analysis and Artificial Intelligence methods. I will illustrate some solutions developed during the 2020 Coronavirus Pandemic for controlling the disease

evolution and for reducing virus spreading. I will describe the first publicly available dataset for the analysis of face-touching behavior that we annotated and distributed, and I will illustrate an extensive evaluation of several computer vision methods applied to the produced dataset. Moreover, I will describe the privacy-preserving solution we developed for estimating the "Social Distance" and its violations, given a single uncalibrated image in unconstrained scenarios. I will conclude the thesis with a Computer Vision solution developed in collaboration with the Egyptian Museum of Turin for digitally unwrapping mummies analyzing their CT scan, to support the archaeologists during mummy analysis and avoiding the devastating and irreversible process of physically unwrapping the bandages for removing amulets and jewels from the body.

Table of contents

Introduction

Introduction

Computer Vision: a field in fast evolution

The history of Artificial Neural Networks applied to Computer Vision is strongly linked with the history of Neuroscience and Artificial Intelligence. It starts back in the 50s and it alternates periods of great enthusiasm, attracting huge investments by government agencies and companies, with periods characterized by difficulties and financial setbacks.

In the 50s, neurology had shown that the brain consists of an electrical network of neurons that fired propagating spikes. Inspired by these observations, in 1957 Rosenblatt proposed the Perceptron [1], an algorithm for learning a binary classifier using simple addition and subtraction. Although the perceptron initially seemed promising, it was quickly proved [2] that perceptrons cannot be trained to recognize many classes of patterns and that they are only capable of learning linearly separable patterns. The perceptron limitations caused the research on Neural Networks to stop for almost 20 years, waiting for greater computer processing power and algorithmic improvements, while the lack of funding slowed down the AI research.

Meanwhile, studying the cat's brain response to visual stimuli, in 1959 the neurophysiologists Hubel and Wiesel [3] showed how the visual system constructs complex representations of visual information from simple stimulus features, through the interaction of simple and complex neurons. Building on Hubel and Wiesel ideas, in 1979 Fukushima proposed the Neocognitron [4], a hierarchical, multilayered Artificial Neural Network used for handwritten character recognition. The Neocognitron is strongly related to the Marr theory, proposed in 1982, which states that the vision process is a hierarchical composition of tasks, starting from a low-level analysis (such as edges and blobs detection) to a high-level understanding of the data [5]. In the same exact way, in Neocognitron, low-level features are composed into larger, more abstract features

In 1974, Werbos created the backpropagation algorithm [6], popularized by Rumelhart, Hinton and Williams [7] in 1986. Backpropagation allows the training of multilayer per-

ceptrons, where many perceptrons are organized into layers and hidden layers are stacked between inputs and outputs. Multilayer perceptrons have greater processing power than Rosenblatt's original perceptrons and they can be used to train deep, highly nonlinear neural architectures. Moreover, as proved by Cybenko's theorem in 1989 [8], multilayer perceptrons are universal function approximators, therefore they can be used to create mathematical models by regression analysis.

Applying backpropagation to Fukushima's architecture, in 1989, LeCun developed the first modern Convolutional Neural Network [9]. Unfortunately, the overestimation of the immediate potential of Artificial Intelligence caused the second stop in funding for AI when the promised results failed to materialize.

Benefitting from the cheap and powerful GPU-based computing systems and the availability of huge amounts of training data easily accessible thanks to Internet, the most recent breakthrough moment happened in 2012 when AlexNet [10] won ImageNet Challenge [11]. ImageNet is a dataset consisting of roughly 1.000 images for each of the 1.000 categories to be discriminated, for a total number of 1.2 millions of training images. A large-scale object recognition challenge has been organized between 2010 and 2017 and the proposed methods were judged by comparing their top-5 classification error rate. In 2012, the Deep Convolutional Neural Network called AlexNet achieved a top-5 error rate of 16%, improving more than 10% the previous year winner performance obtained by using SVMs classifiers. In few years, the ImageNet winners' classification error significantly dropped, surpassing human-level performance in 2015 [12] and reducing the top-5 classification error rate to 2% in the latest challenge. Thanks to ImageNet Challenge, several architectural improvements, algorithmic refinements and practical solutions have been studied and developed in order to overcome the difficulties related to the training of very deep Neural Networks, including the computational and optimization complexities. Among them, it is worth mentioning batch normalization [13], residual connections [14] and architectural search [15].

In few years, Deep Learning has shown to be extremely effective in facing, as never before, an increasing number of complex Computer Vision tasks (such as Image Classification, Image Segmentation [16], Pose Detection [17], Image Captioning [16], Object recognition from Sound [18]) and providing insights in many other AI research fields, such as Natural Language Processing, Recommender Systems, Speech Recognition, and Reinforcement Learning.

Computer Vision research

Given the extremely fast evolution of Computer Vision and Artificial Intelligence fields, also the role of the Researcher is rapidly evolving.

Thanks to the improved results obtained by the development of Deep Learning models, the amount of investigating applications keeps growing: AI is already making an impact on everyone's life, being more and more part of modern society. Vision-based Machine Learning algorithms have been developed for healthcare, automatic industrial inspection, military (unfortunately), drug discovery, agriculture, 3D modeling, assistive driving, sports analytics, information organization and retrieval, movie production, behavior analysis and video surveillance. These applicative scenarios frequently require the development of ad hoc algorithms or the redesign of state-of-the-art methods, therefore Researchers' and Data Scientists' effort is needed for designing suitable solutions. It is not a coincidence that essentially all the most prominent Computer Vision Researchers are collaborating or have been hired by big tech companies (Amazon, Apple, Alphabet\Google, Facebook, Microsoft, Nvidia, Alibaba, Baidu, Tencent, Huawei, IBM, Intel, Toyota, ...). The huge effort spent in solving applicative scenarios, the increasing amount of investments by companies and the growing number of available products embedding AI-based solutions are clear demonstrations of the maturity of the field and of its capability in providing effective answers to social requests.

Despite that, an incredibly big amount of problems related to Deep Neural Networks still remain unsolved, starting from a lack of rigorous mathematical theory surrounding some methods [19] that results in their understanding at a heuristic and empirical level, only [20]. Some of the most relevant topics on which the Scientific community is focusing on are: explainability of Neural Networks [21], robustness to adversarial attacks [22], datasets bias [23], integration with prior knowledge [24], requirement of huge amount of data [25], high-level reasoning and decision making.

Given all these considerations, the Researcher role can be framed in two complementary areas of interest:

- **Methodological research**: theoretical and methodological analysis of architectures and algorithms, to further improve them, develop new ones and identifying their weakness and open issues;
- **Applied Research**: development, customization and redesign of state-of-the-art methods to effectively apply advanced solutions to specific application fields.

Methodological Research: the Big Data tie

The current state-of-the-art Computer Vision technologies are strongly tied to the need for a huge amount of data. Lead by ImageNet dataset availability, containing hundreds of instances per each category, and by the impressive results' improvement obtained on it, supervised learning and discriminative models have flourished and continuously improved. Typical Deep Learning architectures have millions parameters randomly initialized to be trained: AlexNet has 60 millions, VGGNet-16 has 138 millions and also the "lighter" architecture such as GoogleNet has 4 millions and SqueezeNet has 1.25 millions parameters. All these architectures are therefore grounded in the availability of a huge amount of training data.

Unfortunately, in the majority of use cases, data are limited (for instance, biomedical imaging), expensive to collect and label (for instance, industrial inspection) or impossible to acquire due to privacy, safety or ethical issues. In fact, in all the scenarios, we can observe that only a subset of the classes to be modeled contains a large number of samples, whereas most of the remaining ones are sparsely populated [26]. This is, in particular, the case of *abnormalities*, defects or irregularities, which are expected to be, by far, less represented than the normal and regular classes. As a consequence, the collected datasets result to be very unbalanced and the algorithms struggle to properly model all the classes consisting of very few observations.

All these scenarios require ad-hoc methods, either for artificially generate additional training data (using data augmentation or generative networks), either for developing models that can overcome the overfitting issues and learn from a very limited amount of data (few-shot learning, one-shot learning, zero-shot learning).

Big Data and Machine Vision

Detecting defects, scratches and irregularities in a product line is a topic of great interest for a large number of industries. Frequently, the varieties of defects to be detected and localized are limited and relatively homogeneous between instances in comparison to many fine-grained Computer Vision tasks. It is important, therefore, to assess whether there is actually the necessity to deploy Deep Learning methods instead of the classical Machine Learning ones for these applications: in fact, in comparison to Convolutional Neural Networks, the classical methods are less prone to overfitting and they can still provide excellent results when properly tuned.

Industrial scenarios need the development of ad hoc acquisition systems composed of an accurate selection of suitable illuminators, cameras and optics. Particular attention needs to

be paid to details such as image resolution, field of view, depth of field, shutter speed, light pulse and acquisition synchronization, object shadows and reflections elimination. It is not uncommon that more than one imaging\sensory technology is chosen, for instance, RGB, multi-spectral, depth, thermal, audio, in order to combine the benefits of each technology: a bad design of the systems can dramatically compromise all the successive processing steps.

The use of a complex, general-purpose acquisition system during the training stage, only, can reduce the test-time performance variability related to the different hardware choices that are possible? Can we improve the overall machine vision system performance working on the training set, only?

Data labeling is a tedious, time consuming, error-prone activity that frequently requires the collaboration of domain experts. Can we reduce the effort required by the annotators or propagate their activity in more than one image, for free?

Finally, there are many use cases where data is not only scarce but actually not available at all: the distribution of training instances among labels naturally exhibits a long tail and only a few classes contain a large number of samples. In the most extreme situation, the existence of certain classes and their characteristics are known, but no images representing them are available. This problem takes the name of Zero-Shot Learning. It might sound surprising, but humans are particularly good at solving this task: criminals are identified based on their description, rare animals can be spotted and recognized based on their similarity and dissimilarity with known species, stars and constellation can be localized based on their expected shape, color and luminosity, quality inspectors can identify defects they have never seen before. The possibility of achieving human-level results in Zero-Shot Learning has enormous potential in several fields, including machine vision. The difficulties related to this task are mainly the huge class imbalance, the domain shift between distributions of novel and base classes and the necessity to transfer knowledge via multi-modal learning methods [26].

In my Thesis, I will investigate Computer Vision model performance in the context of machine vision and industrial inspection, with particular attention to the difficulties and the costs related to big dataset acquisition. In my study, I will be guided by the following questions:

- Do we need to bother with big data collection and annotation in industrial scenarios?
- How can we maximize system performance minimizing the data labeling effort?

- Is it possible to mitigate the hardware influence on visual inspection system performance?
- Can we design a system that can discriminate against a class that has not been seen in training time?

Applying State-of-the-Art AI for Social Good

As discussed in the previous paragraphs, Machine Learning and Computer Vision went out of the labs and have been deployed into society. The technologies currently available can address and tackle many real-world problems, proposing innovative solutions or improving already existing methods. This emerged even more during the 2020 Coronavirus Pandemic, when Data Scientists have been involved with many other scientists in trying to understand pandemic evolution, to develop solutions for reducing virus spreading, to model the diffusion of the disease, to discover new drugs that can fight the virus and its symptoms.

Actually, many of the biggest social challenges can be addressed with the support of Computer Vision and AI and it has already been done successfully for improving the daily life of people with disabilities, for supporting vulnerable communities, for Cultural Heritage preservation, for mapping the impact of climate change in the planet, for protecting earth biodiversity and for delivering personalized education.

In my Thesis, I will illustrate some of the related solutions I developed during my Ph.D. They represent a selected set of the activities I have been involved. The complete list of my activities and published papers, including a short description of their content, can be found after the Conclusions.

Structure of the Thesis

In the first part of the Thesis, I will investigate some methods for reducing the cost of data acquisition for Deep Learning applications in the relatively constrained scenarios related to industrial inspection. In Chapter 1, I will assess the effectiveness of Deep Neural Networks in comparison with several classical Machine Learning algorithms requiring a smaller amount of data to be trained. Hereafter, in Chapter 2 I will introduce a hardware-based data augmentation approach, which leads to a considerable performance boost taking advantage of a novel illumination setup designed for this purpose. Finally, in Chapter 3, I will investigate the situation in which acquiring a sufficient number of training samples is not possible, in

particular the most extreme situation: zero-shot learning (ZSL), which is the problem of multi-class classification when no training data is available for some of the classes. Visual features designed for image classification and trained offline have been shown to be useful for ZSL to generalize towards classes not seen during training. Nevertheless, I will show that recognition performances on unseen classes can be sharply improved by learning ad-hoc semantic embedding (the pre-defined list of present and absent attributes that represent a class) and visual features, to increase the correlation between the two geometrical spaces and ease the metric learning process for ZSL.

In the second part of the Thesis, I will present some successful applications of state-of-the-art Computer Vision, Data Analysis and Artificial Intelligence methods. I will illustrate some solutions developed during the 2020 Coronavirus Pandemic for controlling the disease evolution and for reducing virus spreading. In Chapter 4, I will discuss about the analysis of face-touching behavior. I will describe the annotated dataset that we publicly distributed and several computer vision methods that we implemented and made available to the scientific community in order to set a baseline performance and promote additional research activities in this relevant topic. In Chapter 5, I will describe the privacy-preserving solution we developed for estimating the "Social Distance" and its violations, given a single uncalibrated image in unconstrained scenarios. In Chapter 6, I will conclude the Thesis with a Computer Vision solution developed in collaboration with the Egyptian Museum of Turin for digitally unwrapping mummies analyzing their CT scan, to support the archaeologists during mummy analysis and avoiding the devastating and irreversible process of physically unwrapping the bandages for removing amulets and jewels from the body.

Part I: Overcoming data scarcity in Deep Learning

Chapter 1

The effectiveness of Deep Neural Networks

1.1 Introduction

Computer Vision systems have been integrated into industrial machines for decades and they currently form an integral part of many production lines. The hardware of a Machine Vision System is typically composed of an illumination system and an acquisition system, which in turn is a combination of digital cameras and optics. These components are mounted in a protecting case that has also the purpose of preserving component reciprocal position. Based on the needs, the system can be designed to accurately analyze either microscopic or macroscopic elements (selecting the proper lens and extender rings), to be dust, water and temperature resistant (by selecting a suitable case), to acquire images frame rate higher than human one (varying chipset and CMOS characteristics), to use specific light bands for highlighting objects' property (using particular illumination setups, suitable camera sensors and optical filters). Optic, light and sensor properties are mutually dependent: for instance, augmenting camera sensor resolution in a production line most of the time requires an increment in the lighting power of the illumination system, due to the smaller sensing area of each physical pixel; similarly, augmenting image depth of field requires the use of an optic with a smaller aperture and an increment in the lighting power of the illumination system. The Machine Vision System hardware is managed by a set of software instructions taking care of the frame acquisition, light/acquisition synchronization, image grabbing, post-processing and all the related subsequent events. The key to Machine Vision System success has been the speed and reliability they have in automatically performing both repetitive and precision-demanding tasks, outperforming human capabilities with a big margin. In

particular, Machine Vision stands out in applications related to quantitative analysis. In fact, the most common classical applicative scenarios for the Machine Vision systems are related to quality inspection, robot guidance and dimensional control. Historically, the algorithms designed to detect defects followed a pipeline which typically involved contrast enhancement, edge linking and refinement [27], [28], [28], [29]. The method proposed in [30] also follows this approach wherein morphological features are used for edge extraction followed by refinement based on curvature. Similarly, the method in [31] uses pixel-neighborhood statistics to identify crack pixels and tensor voting [32] to connect them.

In opposition to quantitative analysis, aesthetic variations and functional problems have been harder to detect by Machine Vision System and humans significantly outperformed industrial systems in these tasks. Recently, the introduction of Convolutional Neural Networks and Deep Learning started to change the trend and expand Machine Vision limits to applications that earlier were not effectively solvable by a machine. In case of defect detection, architectures designed for image classification (inspired from AlexNet [10]) such as [33] and [34] can be used to classify the image patches. Alternatively, The methods proposed in [35], [36], [37], [38], are derived from [39] and follow the approach of labeling every pixel in the image. These networks have a pyramidal form following the U-Net architecture [40]. Additionally, some of them also have side outputs that act as edge priors [41] for defects detection.

In many industrial applications, still, the varieties of defects to be detected and localized are limited and relatively homogeneous between instances, in comparison with many fine-grained Computer Vision tasks. It is important, therefore, to assess if actually there is the necessity to deploy Deep Learning methods instead of the classical Machine Learning ones for these applications: in fact, in comparison to Convolutional Neural Networks, the classical methods are less prone to overfitting and they can still provide excellent results when properly tuned.

1.2 The effectiveness of Deep Neural Networks: An empirical verification

I chose to compare the performance of Convolutional Neural Networks and several Classical Machine Learning algorithms in the use-case scenario of crack detection in ceramic tiles (Fig. 1.1) [1].

Given the chosen acquisition setup [2], images representing 109 defective ceramic tiles have been acquired. For these tiles, ground truth is provided by an expert from our industrial partner in the form of digital annotations which includes cracks' contours and locations of the tile corners for homography based registration (Fig. 1.3). The pipeline used for the experimental procedure is shown in Fig. 1.2. The following subsection explains the components of the pipeline and the extraction of labeled patches from the acquired images and their organization to create the dataset.

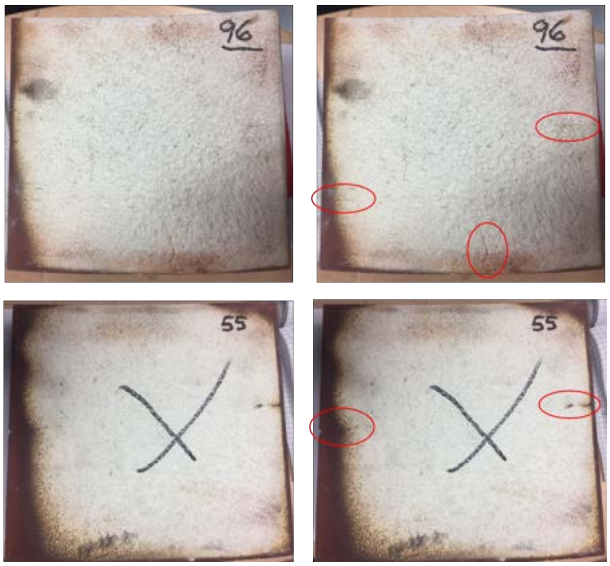


Figure 1.1 Examples of cracks to be detected in ceramic tiles.

1.2.1 Dataset organization

The acquired 640x480 images and the binary ground truth image containing the crack contours are used to extract 50x50 labeled patches in a sliding window fashion with a stride

[1] The results illustrated in this Chapter have been partially included in the paper "*A Versatile Crack Inspection Portable System based on Classifier Ensemble and Controlled Illumination*", ICPR2020 by M. Gajanan Padalkar, C. Beltran-Gonzalez, M. Bustreo, A. Del Bue, V. Murino.

[2] Additional details regarding the acquisition setup cannot be disclosed due to pending NDA agreement.

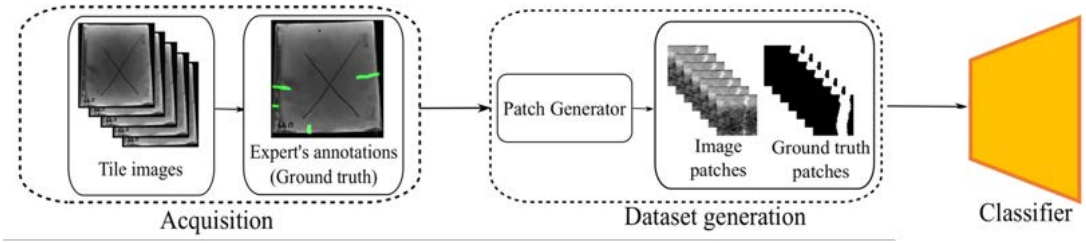


Figure 1.2 Experimental pipeline.

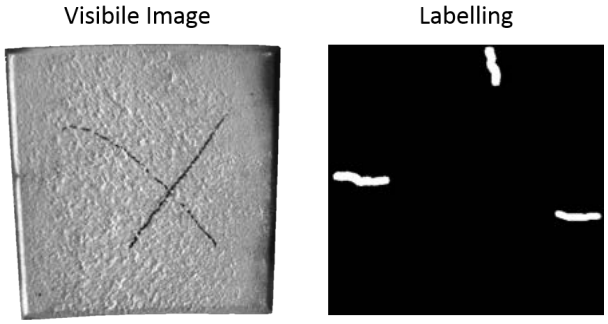


Figure 1.3 Examples of labeling collected from a specific defective tile.

of 10 pixels. During extraction, the image patches are labeled as either positive (patches containing cracks), negative (patches that do not have cracks) or ambiguous based on the proportion of crack pixels in the corresponding ground truth patch (Fig. 1.4). The proportion of crack pixels p in a ground truth patch ψ of size $m \times n$ is defined as:

$$p = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} I(\psi(i,j) > 0) \tag{1.1}$$

where I is an indicator function such that $I(True) = 1$, $I(False) = 0$ and $\psi(i,j)$ is the value of pixel at the location (i,j) in the ground patch ψ.

Using this definition of proportion of crack pixels p, patches in the registered images are labeled as follows:

- Negative patches:$p < 0.1$,
- Ambiguous patches: $0.1 \leq p < 0.2$,
- Positive patches: $p \geq 0.2$.

Only the generated positive and negative patches are used for training and evaluation of the classifiers.

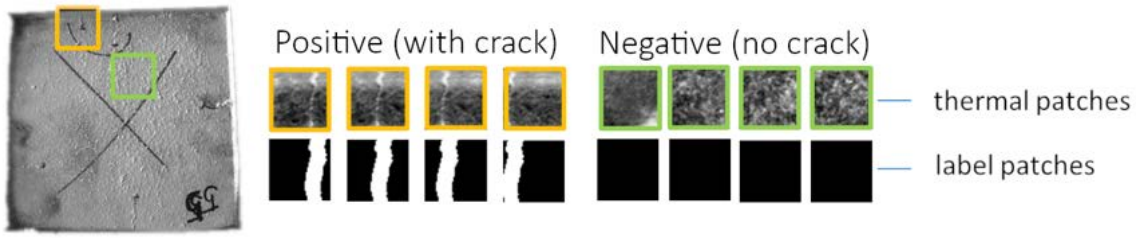


Figure 1.4 Examples of patches extracted from the acquired images and related labeling.

Data Balancing: Since every tile has only a few crack patches, the number of negative (non-defective) examples is substantially larger than the number of positive (defective) examples. Because of that, if we consider all the extracted patches, the resulting Dataset would be unbalanced. If an imbalanced Dataset is used during training, the trained classifier can be biased towards the most represented class, which in our case is the negative class. To avoid this problem, we artificially balance the patches extracted from every image by random undersampling, i.e., we use all the positive patches and randomly select an equal number of negative patches for balancing the dataset.

k-Folds: Given the limited size of the available Dataset, patches extracted from the acquired images are divided into $k = 10$ folds. Using the k^{th} data split as test set and the remaining $k-1$ splits as training set, we trained and evaluated k different models (Fig. 1.5). Using this approach, we can eventually compare models' performance by mean of their average performance and their standard deviation.

All the patches extracted from a specific ceramic tile have been assigned to one and only one fold. This ensures that the patches used for testing in a particular fold are not used for training in the same fold.

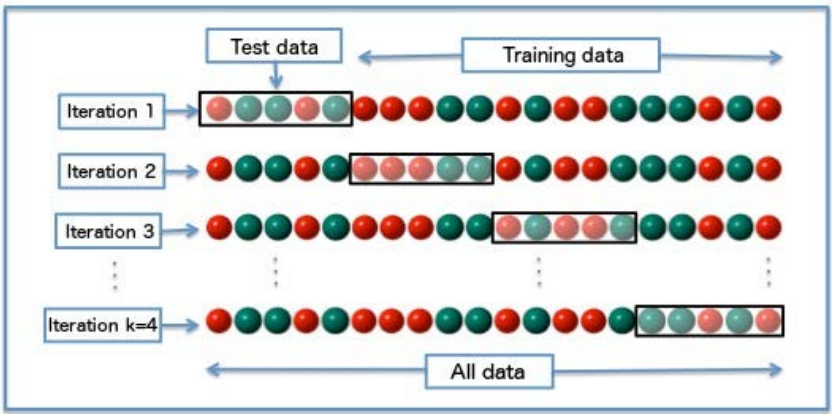


Figure 1.5 Representation of the k-fold cross validation method applied.

Considering data balancing and folds discussed above, we used the following data during the different phases: For fold $= F_K$:

- Train$_K$: {Balanced Positives, Balanced Negatives} from tiles selected for training in F_K;
- Validation$_K$: {Balanced Positives, Balanced Negatives} from tiles selected for testing in F_K;
- Test$_K$: {Imbalanced Positives, Imbalanced Negatives} from tiles selected for testing in F_K.

1.2.2 Evaluation metrics

Since the training is performed on patches, the most straightforward evaluation metric consists of comparing patch labels and output labels. Denoting true positives, false positives, true negatives and false negatives with TP, FP, TN and FN, respectively, the index used for evaluation are the following.

- **Accuracy:** The accuracy represents the percentage of correctly classified patches with respect to the overall number of patches.
  It is defined as follow:
  $$Acc = \frac{TP+TN}{TP+TN+FP+FN}$$
- **F1-score:** F1-score is the harmonic mean of Precision and Recall:
  $$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{1.2}$$
  where:
  - **Precision:** The precision represents the percentage of correctly classified defective patches with respect to the overall number of patches *classified* as defective.
    It is defined as follow:
    $$Precision = \frac{TP}{TP+FP}$$
  - **Recall:** The recall represents the percentage of correctly classified defective patches with respect to the overall number of patches *labeled* as defective.
    It is defined as follow:
    $$Recall = \frac{TP}{TP+FN}$$

- **Matthews Correlation Coefficient:** The previously defined metrics can be misleading in the presence of imbalanced data. Therefore, we also use the The Matthews Correlation Coefficient (MCC) [42], which is a more robust measure in presence of imbalanced data.
  The MCC is defined as follows:

$$MCC = \frac{TPxTN - FPxFN}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}}$$

1.2.3 Classical Machine Learning methods

The complexity of the cracks is not comparable with the most challenging Computer Vision problems currently addressed, nevertheless, due to their variability in shape, depth, length and orientation, learning by example is the most suitable approach to correctly classify the patches.

The model design process can be decomposed into two algorithmic phases: Feature design and Classifier selection.

Feature Design

The feature design phase transforms the input data in a vectorized descriptor. The descriptor should be compact, informative and discriminative. The following descriptors have been tested:

- **LBP: Local Binary Pattern** [43] - LBP algorithm has been originally developed for classifying textures. For each pixel in the image, it compares pixels' intensity with the ones of its 8 neighbors. Given this representation for all the pixels in the image, LBP generates a histogram representing the frequency of each pixel representation. LBP can describe image details at different scales, thanks to the possibility to choose different pixel neighborhood dimensions. To obtain a feature vector with higher discriminative power, we implemented a multi-scale LBP descriptor, concatenating the vectors generated at 3 different scales.

- **HOG: Histogram of Oriented Gradients** [44] - Gradients are a good source of information in an image since they have a high magnitude close to the borders of the framed elements. HOG has been originally developed for person detection. It separates the image in a dense grid of cells and it computes a histogram representing the directions that are present in each cell. The concatenation of the computed histogram

represents the generated feature vector. HOG is a feature representation robust to noise and invariant to light changes;

- **VGGNet FC7 Pre-trained Features** [45] - VGGNet is a Deep Convolutional Neural Network which has been applied for image classification. It is composed of a sequence of convolutional layers. The layers' weights have been tuned during the training stage using ImageNet Dataset. The seventh layer of the network, the latest before the decisional layer, is called FC7. Its output consists of a 4096 elements vector codifying all the information contained in the image (or in the patch, in our case). Features generated at the FC7 level have shown to be able to generalize well between tasks and they can be used as input of a general-purpose classifier.

Classifier Selection

The Classifier should be able to model the statistical relations between different features of the designed descriptors, in order to achieve better accuracy for our model:

- **k-NN: K-nearest neighbors** [46] - k-NN is one of the simplest classification algorithms: it memorizes all the training samples and the associated labels. When a new input vector needs to be classified, the K closest vectors are located using the Euclidean distance, the instances of each labeled class are counted and the most frequent one is associated with the unlabelled input. k-NN is a very powerful non-linear classifier, but it requires a big amount of memory since all the training dataset needs to be memorized. It has the nondesirable feature of being very fast in training but slow at test time since we have to calculate all the distances between the data to be classified and the training set data.

- **SVM: Support Vector Machine** [47] - Using training set vectors and their labels, SVM learns the hyperplane that best separates the classes of the dataset. In case the training data are not linearly separable, SVM can map the inputs in a higher dimensional feature space such that the hyperplane separation of the training data suffices for separating the class instances. The unlabeled inputs to be classified are mapped in the learned hyperplane before classification. SVM has proven to be particularly effective in data-scarce applications and when the dataset is corrupted by several outliers. In comparison to k-NN, SVM is much faster during test and less memory demanding.

- **Random Forest** [48] - The basic building block of the Random Forest is the Binary Decision Trees (BDT). BDT discriminates the training data iteratively looking for the single feature that can better split the data into their labeled classes, using a properly chosen threshold. Random Forest uses multiple BDTs, built from a different subset of the training data, to obtain statistically independent classifiers. Random Forest is very fast in test time.

All the mentioned classifiers receive a vector as input, therefore they can be fed by anyone of the selected descriptors. Both feature design and classifier selection impact the performance of the final model and the best descriptor-classifier combination is application dependent, therefore it is necessary to exhaustively experiment all the 12 possible combinations to assess the best one.

Results

For better clarity, we present the results organized by classifier, comparing the performance obtained varying the input features for each of them: k-NN (Fig. 1.6), SVM (Fig. 1.7), Random Forest (Fig. 1.8). In each experiment, the 10 chosen folds produced similar results, as it can be noticed by the plots, where the central line represents the median value, the margins of each box represent the range between 1^{st} and 3^{rd} quartile and the whisker report the maximum and minimum obtained values.

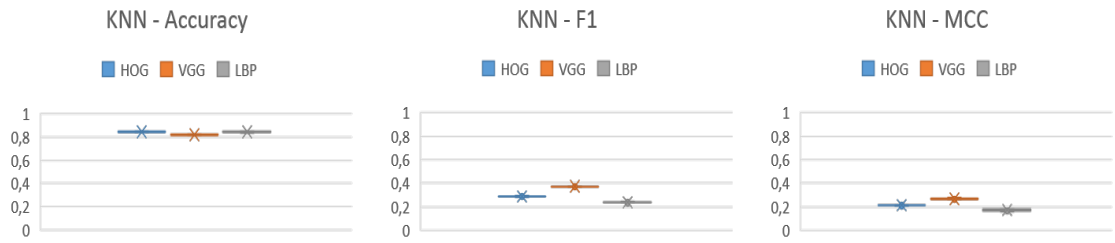


Figure 1.6 Crack detection performance varying input features and using k-NN as classifier.

The graphs show that, independently from the chosen classifier, the best performing descriptor is VGG-FC7, therefore, in Fig. 1.9 we compare classifiers' performance, using VGG-FC7 as input feature.

We observe that the performances obtained with the different chosen classifiers are very similar: accuracy is slightly over 0.8, F1-score is close to 0.4 and MCC is close to 0.3. In all the considered metrics, VGG-FC7 and SVM is the best performing combination.

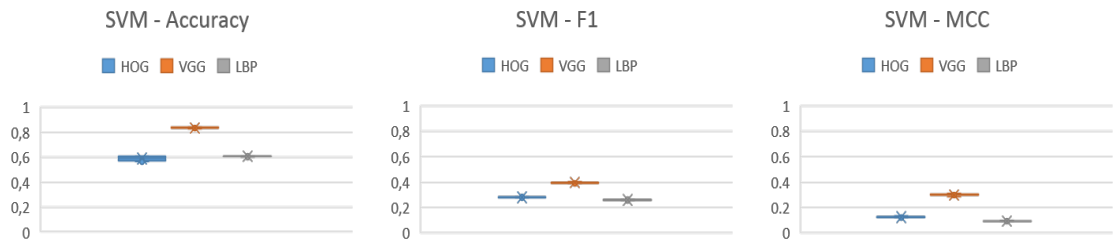


Figure 1.7 Crack detection performance varying input features and using SVM as classifier.

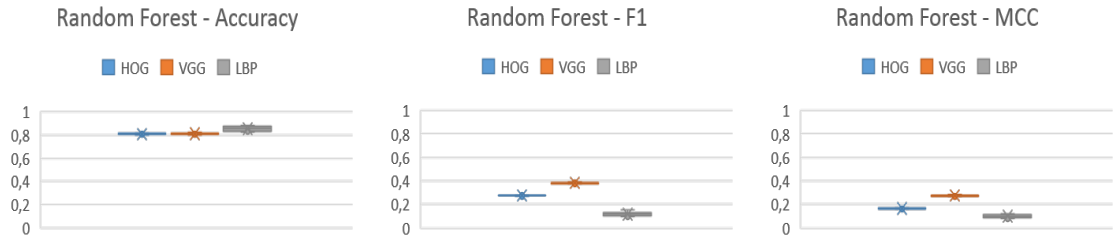


Figure 1.8 Crack detection performance varying input features and using Random Forest as classifier.

1.2.4 Convolutional Neural Networks

The results described in Section 1.2.3 highlight the effectiveness of ImageNet pre-trained VGG features. In this Paragraph, we will assess end-to-end CNN training performance and compare it to the best combination of feature and classifier analyzed in the previous Paragraph. Considering the effort required to produce an exhaustive assessment of the classical Machine Learning methods and an appropriate hyper-parameter tuning for all the possible combinations, the output of the comparison is extremely valuable for companies and universities having to quickly select the most promising architecture or method.

In Fig. 1.10 we showed the chosen architecture for the Convolutional Neural Network, inspired by LeNet-5 [49] and VGG [45], designed and trained for classifying tile patches as either defective or non-defective.

The choosen Convolutional Neural Network consists in:

$$\begin{gathered}\text{Conv 5x5x16} \rightarrow \text{ReLu} \rightarrow \text{MaxPooling} \rightarrow \\ \text{Conv 3x3x32} \rightarrow \text{ReLu} \rightarrow \text{MaxPooling} \rightarrow \\ \text{Conv 3x3x64} \rightarrow \text{ReLu} \rightarrow \text{MaxPooling} \rightarrow \\ \text{Fully Conected (256 hidden units)} \rightarrow \text{ReLu} \rightarrow \\ \text{Fully Conected (256 hidden units)} \rightarrow \text{ReLu} \rightarrow \\ \text{Fully Conected (2 hidden units)}\end{gathered}$$

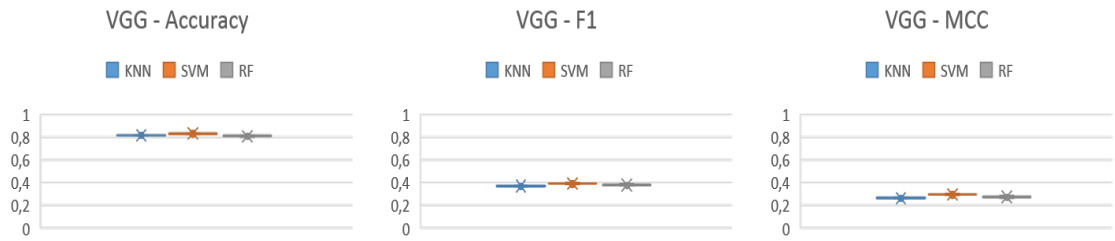


Figure 1.9 Comparison of crack detection performance using VGG-FC7 as input feature and varying the classifier.

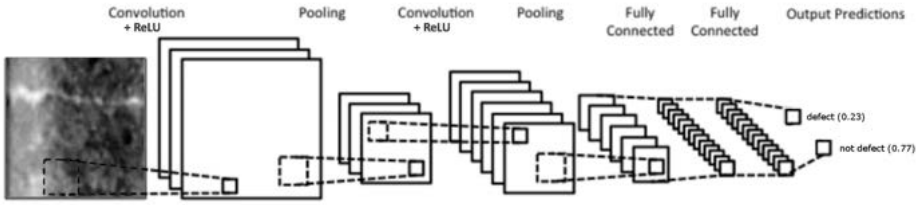


Figure 1.10 Convolutional Neural Network trained for crack detection application.

In Fig. 1.11 we compare CNN performance with the ones obtained in the previous Paragraph using VGG-FC7 as input feature.

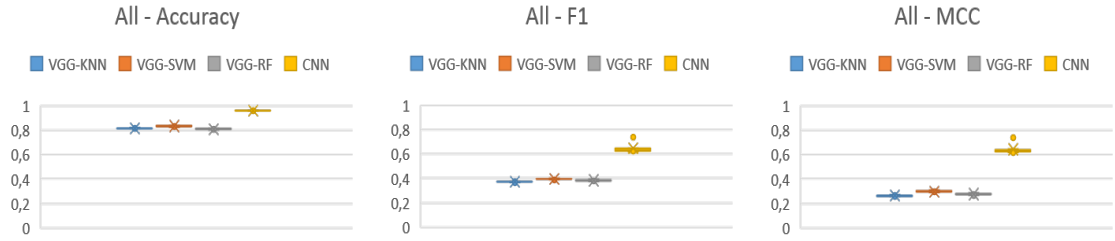


Figure 1.11 Comparison of crack detection performance using CNN architecture and feature/classifier solution based on VGG-FC7 input feature.

It can be easily observed that the CNN architecture, despite being relatively shallow, significantly outperforms all the previously tested methods in all the considered metrics.

1.3 Conclusions

The experimental results presented in the current Chapter justify the most recent trends of the Computer Vision community, which is gradually abandoning the classical Machine Learning methods preferring Neural Network based approaches.

The established methodological procedures developed in the last decades include the use of data augmentation, suitable weight initialization (such as Adam), appropriate non-linearities choice (such as ReLu), adoption of batch normalization and dropout. All of them and the recent hardware improvements (such as the availability of GPUs and TPUs) make Neural Network based solutions the current most promising approaches when dealing with image classification tasks.

Chapter 2

Reducing the labeling effort: Hardware-based data augmentation

2.1 Introduction

In Chapter 1, we illustrated the effectiveness of Neural Networks based solutions over classical Machine Learning methods in image classification applied to visual inspection.

In practical applications, the biggest drawback of Deep Neural Networks (DNNs) is the requirement of a huge amount of training data. The required size of the training set increases with the complexity of the problem to be solved and with the size of the architecture used for solving it. Having a large dataset is fundamental for the performance of the deep learning model. Just for having a reference order, the already mentioned AlexNet architecture [10], winner of ImageNet Large Scale Visual Recognition Challenge 2012 [11], has 60 million parameters and it has been trained using ImageNet [50] training set, consisting of roughly 1.000 images for each of the 1.000 categories to be discriminated, for a total number of 1.2 millions training images.

Currently, Deep Learning works best when there are thousands, millions, or even billions of training examples [19]. This is due to statistical assumptions, architectural design choices and problem-specific characteristics. In fact, the modern big data training goal is modeling the dataset's underlying statistical distribution looking for the minimum in an error surface define by the Neural Networks parameters. The assumption is that the error surface is smooth and that the minimum can be gradually reached using gradient descend. Deep Neural Networks have a huge number of tunable parameters, therefore the error surface is particularly complex and non-convex, requiring a big amount of data to successfully search for the optimal point, in particular if it is initialized randomly.

Furthermore, in case the training dataset is not big enough, it won't be representative of all the possible real-world instances, and Convolutional Neural Networks have difficulties in generalizing to novel viewpoints and to transformations that are different from translation [51].

In case the training dataset is not big enough and it does not suffice for properly training the Neural Network, we fall into the problem known as over-fitting: since the training dataset is not representative of all the possible real-world instances, the trained Neural Networks will have difficulties in generalizing to novel instances that are different from the seen ones.

For industrial applications, the collection of annotated data is particularly challenging, expensive and time-consuming. Data acquisition requires the development of a suitable vision system and high-quality labeling requires the support of persons with strong experience in the domain application field. In this Chapter, we will present a hardware-based solution that can mitigate the labeling effort required for acquiring a big training dataset. We will show that proper light variations during image acquisition lead to complementary information that the Neural Network architectures are able to exploit during training for better solving the objective goal. The deployment of the proposed hardware during both training and testing phase provides the most accurate results, nevertheless, the substitution of the acquisition system is not feasible in most of the real-world industrial scenarios. We will demonstrate that using the proposed hardware during the training stage, only, is sufficient to improve Neural Network models performance in comparison to the baseline.

We tested the effectiveness of the idea in an eyewear quality inspection scenario. This scenario is particularly challenging, in fact, for any given manufactured eyewear, countless different models might exist, varying in material properties (specular, diffusive, directional, transparent) or in geometrical shape (flat, curved, prismatic). The surface to be inspected might also contain patterns and adornments which should be distinguished from the undesirable irregularities (see Fig. 2.1)

2.2 The standard approach: software-based data augmentation

The basic and most straightforward approach used by the community for mitigating the overfitting problem deriving from limited dataset availability is augmenting the data already

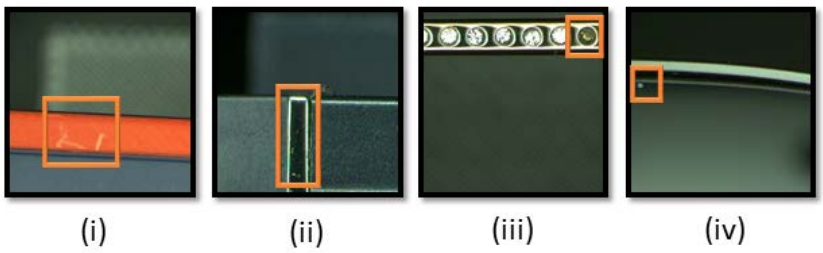


Figure 2.1 Four examples of defective images in our dataset: (i) worn-out paint, (ii) dots on the metal surface, (iii) missing decorations, (iv) unexpected glass break.

available through suitable software transformations. The most common transformations used for data augmentation are horizontal and vertical flipping, random cropping, hue and saturation adjustment, affine and perspective transformations, Gaussian noise, blurring (Fig. 2.2).

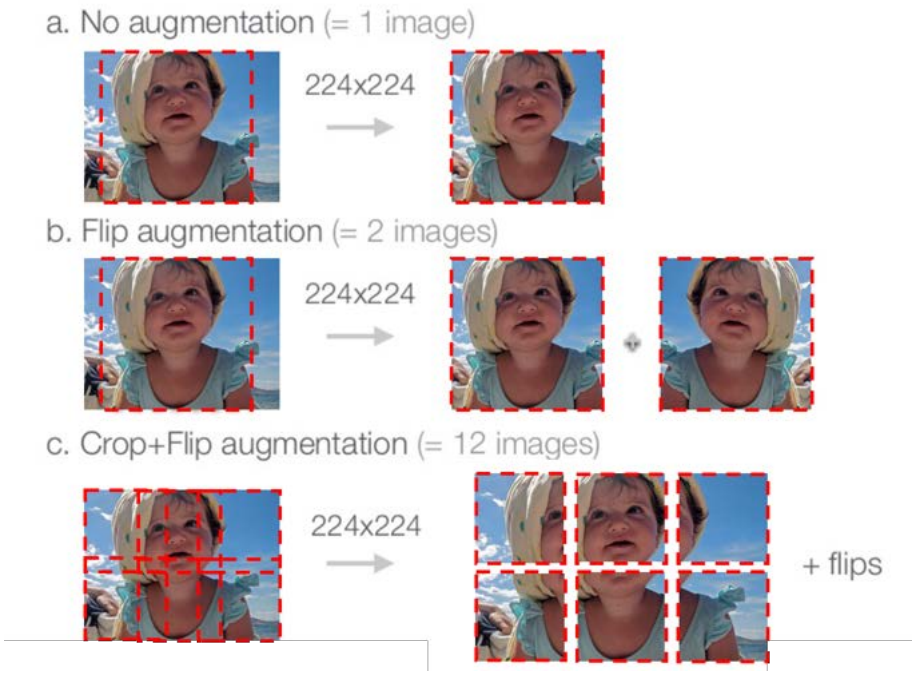


Figure 2.2 Software based data augmentation.

Data augmentation is a technique used for enriching the representativeness of the available training dataset with the goal of better modeling the underlying data distribution. Every time the same image is analyzed by the algorithm during training, a combination of the chosen transformations is applied to it. We decided to study the possibility to develop a hardware-based data augmentation to be used side-by-side with the most common software-based one.

The key idea for achieving the mentioned goal has been developing a novel illumination setup capable to concurrently acquire multiple images of the same object, sharing the same labeling. The novel lighting setup embeds four interchangeable illumination configurations while preserving the camera-object reciprocal position [1].

2.3 The proposed approach: hardware-based data augmentation

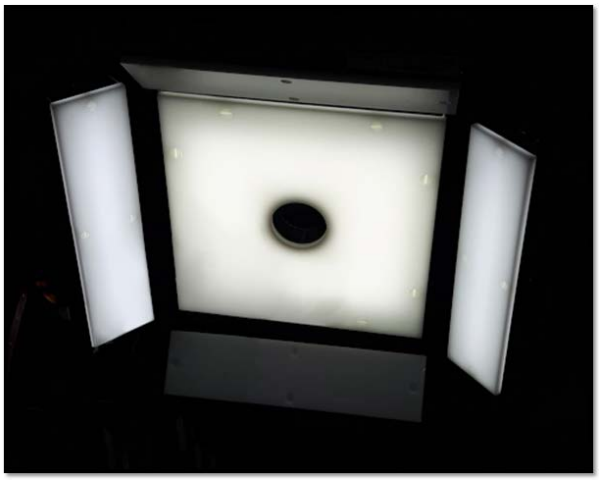


Figure 2.3 The proposed lighting system.

2.3.1 Hardware setup definition

A standard visual inspection hardware setup is typically composed of a digital camera, optics and an illumination system. A customized software evaluates the captured images and eventually takes decisions based on the results of the evaluation. Hence, hardware selection is a fundamental task in the design of an automatic visual inspection system and it is essentially driven by the characteristics of the object to be inspected [52].

We designed and developed a lighting setup composed of five flat-dome lights that alternatively activate and deactivate in different combinations. Dome light offers diffused, shadow-less, and uniform illumination in complex surfaces, including shiny, curved, and

[1] The results illustrated in this Chapter have been partially included in the paper "*Complex-Object Visual Inspection: Empirical Studies on A Multiple Lighting Solution*", ICPR2020 by M. Aghaei, M. Bustreo, P. Morerio, N. Carissimi, A. Del Bue, V. Murino.

uneven ones. Flat-dome lights provide the same characteristics as dome lights, with the additional advantage of occupying a smaller volume, comparable with the one occupied by standard LED lights. The light positioning has been empirically studied to reproduce diffused, dark-field and frontal lighting techniques, while producing the least possible glares on the specular surfaces. To further minimize the reflectivity of the lighting system, which would make it visible when acquiring highly specular surfaces, we covered all the white flat-dome lights with dark collimator filters. Our proposed setup can be seen in Fig. 2.3 and in Fig. 2.4.

We identified four lighting configurations which allow the system to produce diffuse front lighting (Fig. 2.4.C) and dark-field lighting from vertical (Fig. 2.4.UD), horizontal (Fig. 2.4.LR) and all lateral (Fig. 2.4.UDLR) directions. Front lighting is mostly suitable for detecting color irregularities or flat defects, while dark-field lighting is extremely useful for acquiring effective images of defects related with surface irregularities such as scratches, bumps, or missing pieces.

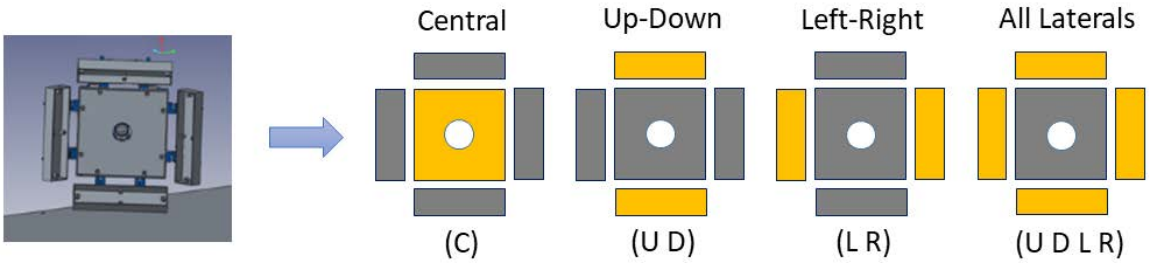


Figure 2.4 Illuminators in the proposed acquisition setup are set to activate and deactivate sequentially, to resemble diffused, dark-field, and front lighting techniques within four illumination configurations (C, UD, LR, UDLR).

In addition to the four lighting configurations, to ensure the appropriate illumination level in the acquired images, independently from the object surface reflective characteristics, each light configuration can be activated for 3 different time lengths (exposures), mimicking 3 different camera shutter speeds (low, medium, high). Camera exposure time is set to be constant and longer than the maximum time of the light activation. Trigger controls are configured such that the lights and the camera are properly synchronized. In our study, all the images are acquired using a Basler acA2440-75uc camera and an Edmund Optics 16mm F1.4 lens. The camera is placed in an ad hoc hole at the center of the central light. In order to block out all the external environmental lights, the entire setup and the object to be inspected has been placed in a dark black box.

In the following of the Chapter, we will refer to *Illumination Configuration* when referring to one of the four possible activation patterns of the lighting system (Fig. 2.4) and to *Illumination Condition* when referring to one of the 12 possible combinations of illumination configuration and exposure (Fig. 2.5).

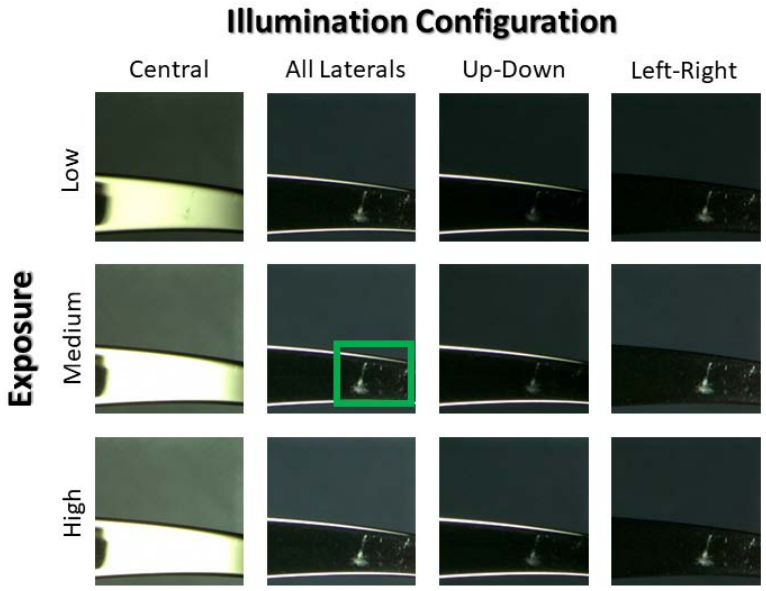


Figure 2.5 Images of a defective object under all the available illumination conditions. The labeled defect annotation by the annotator is shown with a green bounding-box.

2.3.2 Dataset collection and organization

Given the acquisition setup described in Section 2.3.1, the system can simultaneously acquire 12 images of the same object, varying only the *Illumination Conditions* (4 *Illumination Configurations*, each with 3 exposures). A defect, depending on its type and depending on the characteristics of the surface where it appears, might be visible in all or only in some of the 12 captured images. For example, as in the case shown in Fig. 2.5, the defect is visible in all the images but images captured with central light with medium and high exposures. Note the significantly different representation offered by each one of the light configurations, despite the invariance of the framed scene.

For each defective object, the annotators label the defect in only one of the images on which they can spot it, as shown by a green bounding-box in Fig. 2.5. No predefined instructions on image choice have been given to the annotators about choosing one illumination condition or another. The normalized frequency of annotations for each illumination condition can be analyzed in Fig. 2.6. Since the framed scene is not varying, a defect is either

present or absent in all the 12 acquired images. Because of that, we applied the collected annotations to all the 11 images corresponding to the remaining *Illumination Conditions*. If the existing defect on the object is not visible in any of the 12 images captured by the setup, the annotator indicates the non-visibility of the defect using a properly defined box in the annotation tool. It is worth mentioning that the developed setup enabled to visualize and correctly annotate 99.2% of the defects in a freely selected collection of objects.

The collected dataset consists of 5.071 defective regions. For each region, 12 images with different *Illumination Conditions* are collected, obtaining a total number of 60.852 images. For our experiments, we split the dataset (or its portions, as described in the Section 2.3.3) in training, validation, and test set with the ratio of 70%, 15%, and 15% respectively.

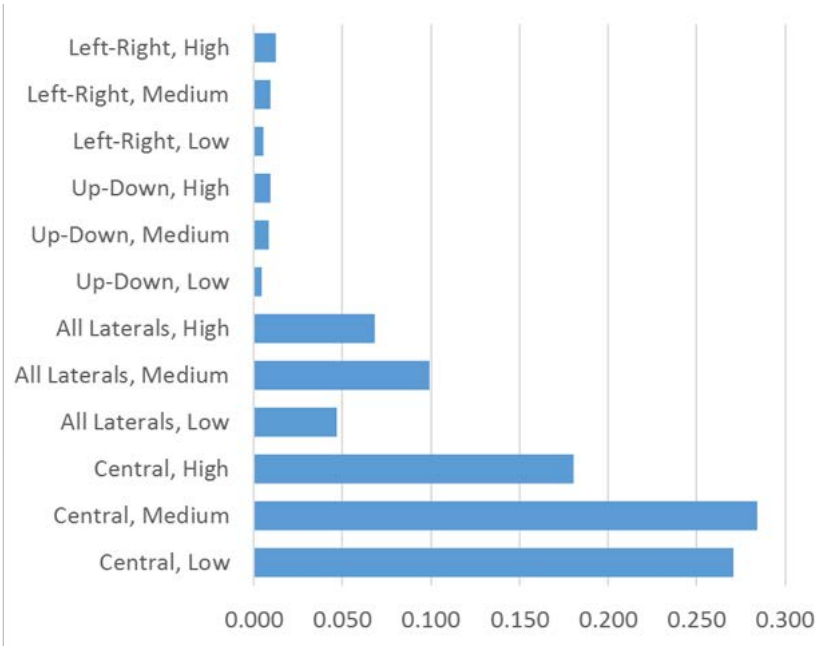


Figure 2.6 Normalized annotations frequency per illumination configuration.

2.3.3 Methodology

In the most common real-world industrial scenarios, the illumination setup is identical during both training and evaluation phase and frequently it has already been chosen and deployed. Changing an already deployed hardware would require massive investments in terms of costs, time and system redesign. The custom illumination setup we developed will only be used during the training phase. The only assumption we make is that the already deployed illumination can be reproduced with one of the *Illumination Conditions* we can obtain using the custom illumination setup described in Section 2.3.1.

We will set up the following experiments to set the baseline performance and to evaluate the improvements that can be achieved thanks to the development of our proposed setup. Fig. 2.7 represents the different training set used for our Studies, as detailed explained in the following Sub-sections.

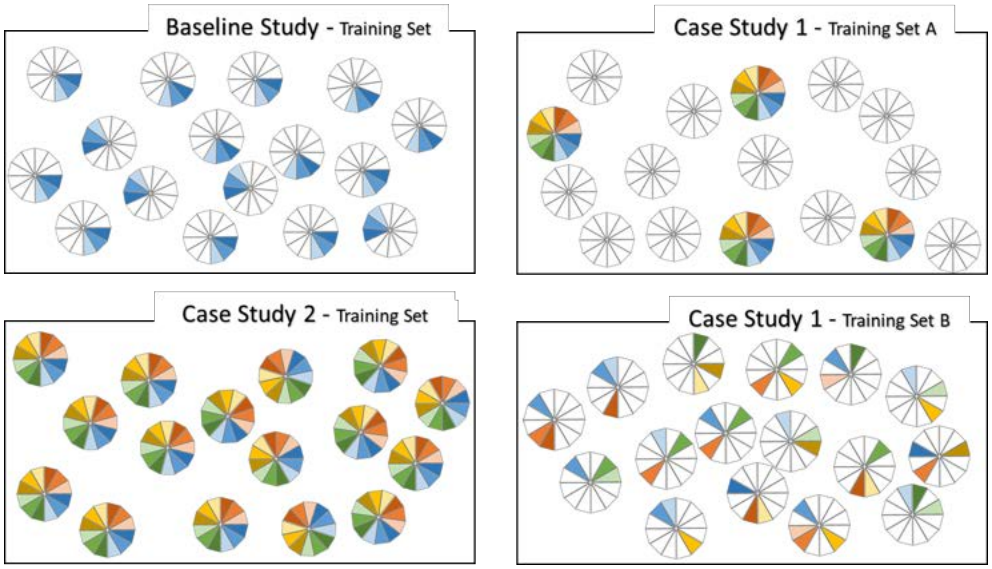


Figure 2.7 Graphical illustration of the different training sets used in our Studies. Each defective sample is represented by a dodecagon. The images collected with a specific *Illumination Conditions* are represented by 12 slices having different colors. They are grouped in the 4 *Illumination Configurations* represented in shadows of blue, green, yellow and orange. The white slices represent discarded images. In Case Study 2, the Training Set is 4 times bigger than the other training sets.

Baseline Study: Training and evaluation using the same single Illumination Configuration

With this set of experiments, we will test the four *Illumination Configurations* that can be obtained with our custom hardware, in order to detect the one which is performing better in our test-case. This set of experiments will be our baseline and represents the most common industrial scenario: the same illumination configuration is available for both training and evaluation.

Note that, in this Baseline Study, only one quarter of the collected dataset is used, since the images related to 3 out of 4 *Illumination Configurations* are discarded. Yet, in all the experiments, the selected *Illumination Configurations* includes all of its corresponding images taken under all the 3 exposures.

Case Study 1: Training using multiple Illumination Conditions and evaluation with single Illumination Configurations.

As already mentioned, in many practical cases visual inspection systems cannot be arbitrarily chosen or modified in the evaluation phase. In this Study, we aim to verify whether acquiring images using multiple *Illumination Conditions* during the model training phase, only, can lead to improved performances on the unmodified single *Illumination Configurations* evaluation setup.

In order to be comparable with the results of the Baseline Study, we will not modify the number of images used during training, i.e. we will continue using one quarter of the entire dataset, only. As described in Section 2.3.2, we have $N = 5.071$ defective regions available, described by 12 images each. Therefore, two possible strategies can be chosen for selecting the training set:

- **Training Set A:** Training Set A is composed by $\frac{N}{4}$ defective regions, described by 12 images each;
- **Training Set B:** Training Set B is composed by N defective regions, described by 3 randomly selected images from the 12 available.

Comparing the performances obtained by training the model on the Training Set A and on the Training Set B will give us a disentangled insight on the comparative effectiveness of both *Illumination Conditions* and defects availability in system performance.

Case Study 2: Training using all the available Illumination Configurations and evaluation with single Illumination Configurations.

Our proposed acquisition setup enables collecting 12 images per each object with no additional effort required for either acquiring and annotating them. In comparison to the previous experiments, In this Study, we will use the entire training set introduced in Section 2.3.2 which is four times bigger than the ones used in the Baseline Study and in Case Study 1.

With this Study, we will verify if the additionally collected images provide beneficial information for training a more effective model to be deployed in a single *Illumination Configuration* scenario, i.e. if the trained model is able to transfer the information collected from one *Illumination Configuration* to a different one.

2.3.4 Experimental setup

We used YOLO-v3 end-to-end detection pipeline [53] in all the discussed experiments, given its fast inference time and its ability to detect small defects. A comparative study of detection algorithms is out of the scope of this Thesis.

YOLO-v3 detector has been originally trained over the COCO dataset [54], then the weights of the network have been adapted to our task using the transfer learning approach updating all the layers of the network.

As mentioned in Section 2.3.2, the dataset is split into training, validation, and test sets. In the experiments where a subset of the training set is required (Baseline Study and Study 1), it has been selected within training, validation, and test sets independently. The splits did not vary in the experiments belonging to the same Study, or shared among various Studies, to preserve comparability. The validation set is used to tune the parameters of the algorithm and the final results are reported on the test set. Each detection bounding-box proposed by the model is compared with the ground-truth and classified as:

- True Positive (TP): the detection has IoU $\geq Threshold_{IoU} = 0.5$ with labeled ground truth and it is therefore considered correct;
- False Positive (FP): the detection has IoU $< Threshold_{IoU} = 0.5$ with labeled ground truth and it is therefore considered wrong;
- False Negative (FN): the labeled ground-truth annotation has not been detected.

We report the results of all experiments using the standard metrics used in single-object (defect) detection as Precision, Recall, F1-score introduced in 1.2.2. Since we can vary the aforementioned metrics by adjusting the acceptance confidence $Threshold_{DET}$ of the detection algorithm, in this Chapter we will also use the Average Precision (AP) metric. AP summarize in a single value the Precision-Recall curve generated varying $Threshold_{DET}$. Being P_t and R_t the Precision and Recall at the t^{th} threshold, we calculated the Average Precision as:

$$AP = \sum_t (R_t - R_{t-1}) P_t \tag{2.1}$$

To compare the results in the next Sections, we will mainly refer to AP, since it considers Precision and Recall relations more globally than F1-score [55].

Training has been done on a NVIDIA GeForce RTX 2080 Ti GPU, with learning-rate = 0.0001, and momentum = 0.9. The hyperparameters have been chosen based on the best performance obtained in the validation set using Random Search [56].

2.4 Results

Baseline Study: Training and evaluation using the same single Illumination Configuration

The results of the Baseline Study discussed in Section 2.3.3 are given in Table 2.1. The most effective *Illumination Configuration* according to the AP and F1-score is the one activating all the lateral lights to produce dark-field illumination from four directions (Fig. 2.4.UDLR). This configuration outperforms frontal light and dark-field illuminations in both vertical and horizontal directions.

Table 2.1 Results of Study 1

Train	Test	Precision	Recall	F1-score	AP
C	C	63.53	45.84	53.25	29.97
U D	U D	61.69	44.95	52.01	29.11
L R	L R	58.56	41.07	48.28	25.52
U D L R	U D L R	61.06	52.73	**56.82**	**34.69**

Case Study 1: Training using multiple Illumination Conditions and evaluation with single Illumination Configurations.

The results of the Case Study 1, discussed in Section 2.3.3, are reported in Table 2.2. Each training set has been generated 5 different random times for each experiment and the AP results are given in $mean \pm std$ format. Precision, Recall, and F1-score values are given for only the first trial.

Table 2.2 Results of Study 2

Train ($N = 5$)	Test	Precision	Recall	F1-score	AP
Training Set A	C	64.86	49.38	56.07	33.18±1.5
Training Set B		66.28	38.00	48.30	25.74±2.75
Training Set A	U D	64.74	51.00	57.06	33.29±1.3
Training Set B		66.98	36.94	47.62	27.17±3.9
Training Set A	L R	65.01	51.37	**57.39**	**34.49±1.57**
Training Set B		68.48	36.90	47.96	27.45±2.9
Training Set A	U D L R	63.53	48.32	54.89	31.84±0.65
Training Set B		66.34	36.61	47.18	26.87±3.79

The results provide an insight that might have been unexpected, a priori: fixed the number of training set images, increasing the heterogeneity of the *Illumination Conditions* is more effective than acquiring more samples of defective objects with a limited set of *Illumination Conditions*. Comparing the results of Study 1 with the Baseline Study, it is noticeable that training with multiple *Illumination Conditions* is beneficial for most of the deployed single *Illumination Conditions*, increasing test performance for all of them except *UDLR*.

Case Study 2

The results of Case Study 2, discussed in Section 2.3.3, are listed in Table 2.3. Using a bigger training set leads to a considerable performance boost (at least $\sim 18\%$) in comparison to the results shown in Table 2.2.

Table 2.3 Results of Study 3

Train	Test	Precision	Recall	F1-score	AP
All Train	C	72.61	70.23	71.39	52.29
	U D	70.69	71.22	70.95	52.27
	L R	73.76	68.87	71.23	**52.57**
	U D L R	72.11	70.37	71.23	52.38

These results are a clear demonstration that acquiring more images thanks to our proposed hardware setup is actually enriching the information provided to the model during training. Even if 3 *Illumination Configurations* out of 4 are not used during evaluation, their availability during training enables the model to perform significantly better in the detection task to be solved. Moreover, it is worth noting that training the model using multiple *Illumination Configurations* reduces and almost remove the difference in performance varying test time *Illumination Configuration*.

2.5 Discussion and conclusions

In this Chapter, we introduced a custom-designed illumination system which is able to mimic four standard illumination techniques: diffused, dark-field, lateral and frontal illumination.

Since the deployment of the proposed setup might not be feasible in inspection environments due to the costs and the technical difficulties related to the hardware substitution, we conducted three Studies to exploit the role of each of *Illumination Configuration* and to analyze whether it is possible to improve system performance when it is deployed in the training phase, only. The conclusions from the studies can be summarized as follows:

- In the case of deployment of the same single *Illumination Configuration* in both training and evaluation phase, the most effective one has been found to be activating all the lateral lights, resembling dark-field illumination from four directions;
- Given the same number of images in the training set but acquiring them using multiple *Illumination Configurations*, the evaluation results are less dependent on the type of *Illumination Configurations* chosen in evaluation phase;
- Increasing the number of training samples using all the available *Illumination Conditions* brings to a large improvement when evaluated on a single *Illumination Configurations*, justifying our proposed lighting setup to be employed at least for training purposes

Chapter 3

Reducing the need for data: Zero Shot Learning

3.1 Introduction

In Chapter 1, I assessed the effectiveness of Deep Neural Networks in comparison with several classical Machine Learning algorithms requiring a smaller amount of data to be trained. In Chapter 2, I showed that the effort required for data collection and labeling can be limited when suitable acquisition systems are studied and developed.

Nevertheless, in many situations is not possible to acquire the required amount of training data and, in all the scenarios, we can observe that only a subset of the classes to be modeled contains a large number of samples, whereas most of the remaining ones are sparsely populated [26]. In the most extreme situation, the existence of certain classes and their characteristics are known, but no images representing them are available. This problem takes the name of Zero-Shot Learning. [1].

Zero-shot learning (ZSL) is the problem of multi-class classification when no training data is available for some of the classes. Precisely, this is the case of *inductive* ZSL as opposed to (the easier) *transductive* ZSL case, in which un-annotated instances from the test classes are used for training. Being motivated by the well known "long tail distribution" [57],

[1] This Chapter has been published as: "*Enhancing Visual Embeddings through Weakly Supervised Captioning for Zero-Shot Learning*", IEEE International Conference on Computer Vision Workshops (ICCVw), 1st International Workshop on Multi-Discipline Approach for Learning Concepts - Zero-Shot, One-Shot, Few-Shot and Beyond, 2019 by M. Bustreo, J. Cavazza, V. Murino.

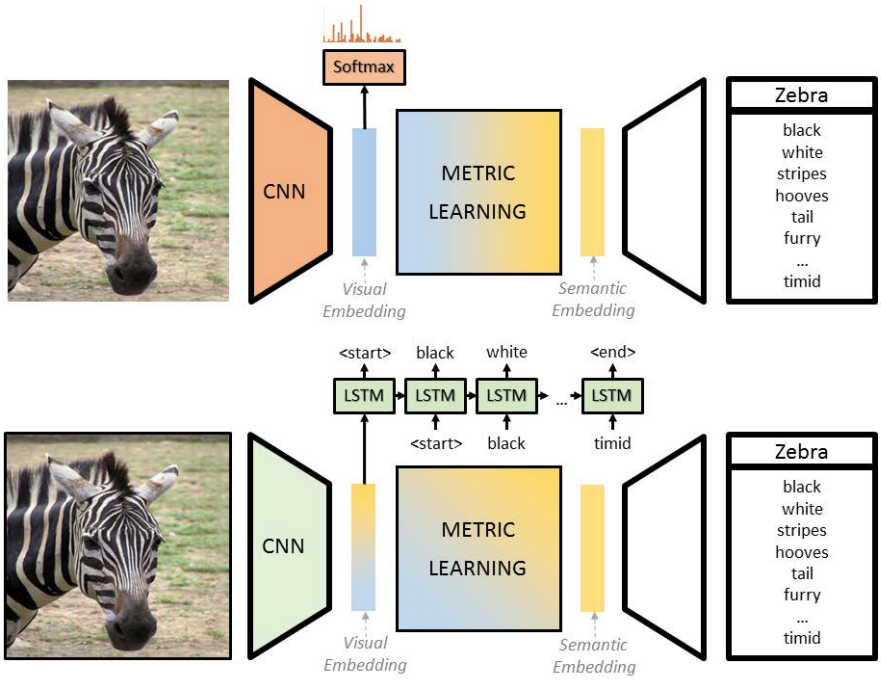


Figure 3.1 *Top*. In Zero-shot learning, class-related attributes are matched with visual features, the latter being usually extracted from a CNN trained for classification (represented in orange). *Bottom*. Differently, to get visual embeddings for ZSL, we exploit the CNN encoder (represented in green) of a CNN+LSTM captioner which predicts attributes at the image level. Our enhanced visual embeddings (*VisEn*) can replace default ones without modifications in the metric learning pipeline. Since containing more semantic patterns (see Fig. 3.3 and Fig. 3.4), *VisEn* is capable of boosting ZSL recognition performance (see Section 3.4).

ZSL has recently attracted a vibrant interest in the computer vision community (see [58] for a survey). In order to recognize *unseen* test classes, ZSL typically leverages auxiliary information, such as attributes, which is required to be both discriminative and shared with the *seen* training classes.

The seminal papers [59; 60] first solved zero-shot recognition (for image classification) by jointly predicting attributes. Recently, attributes prediction was replaced by *metric learning* [61; 62; 63; 64; 65; 66; 67; 68; 69; 70; 71] to implicitly infer the degree of compatibility between *semantic embeddings* - encoding attributes at the class level - and *visual embeddings* which encodes images.

Circumventing attributes prediction in ZSL is practically reasonable. In fact, to reliably predict attributes in the zero-shot setup, modern approaches exploit natural language processing and captioning techniques, both requiring a strong level of supervision. In fact, in *zero-shot captioning* [72; 73; 74; 75; 76; 77; 78], each training instance is annotated by

humans with multiple detailed descriptions (*e.g.*, 15 each in [72]). Such wealth of annotations is unfortunately not available in ZSL benchmarks, where, instead, attributes are annotated at the instance level by registering the presence or absence of attributes within a pre-determined list (as in CUB [79]). Moreover, in some cases, attributes are annotated at the class level only (as in AWA2 [80]) by providing a measure of coherence between each attribute and each class. This makes semantic embeddings **not** *visually grounded*: *e.g.*, we can expect a strong semantic coherence between the attribute "quadrupedal" and the class "zebra". But, crucially, such information can fool a ZSL model when recognizing an image of a zebra, only depicting its upper body, whose legs are not visible (as in Fig. 3.1).

In order to tackle this problem, we propose to enhance the semantic content of visual embeddings by extracting them from the CNN image encoder of a LSTM captioner that predicts attributes. To do so, we convert a captioner to operate in the weakly supervised regime which is common to ZSL benchmarks. That is, we do not take advantage of several natural sentences describing each instance as in [72; 73; 74; 75; 76; 77; 78]. Differently, we only rely on attributes annotated either at image-level or at class-level. In the latter case, we generate for free image-level supervision by leveraging the following observation: if an attribute is semantically incompatible with a given class, then, all instances of that class will not show that attribute as well. For instance, because zebras do not fly, we can bet on the fact that, within any (realistic) image of a zebra, wings won't be present. Hence, we train our captioner to predict which attributes are missing at the image level.

As opposed to default visual embeddings designed for classification, we posit that our captioner-based *enhancement* is capable of enriching visual features of semantic content. Consequently, ZSL is expected to be eased since our enhanced visual embeddings (termed *VisEn*) are designed to convey visually-grounded semantic cues, whereas default visual embeddings are not.

Through a broad experimental validation, we assess the capability of *VisEn* in capturing semantic patterns by evaluating attribute prediction both qualitatively and quantitatively. Further, through an ablation study, we show *VisEn* to be capable of:

1. Being superior to classically adopted visual embeddings (*i.e.*, GoogleNet or ResNet-101 features);

2. Boosting in performance existing ZSL methods.

In practical terms, *VisEn* is compatible with any generic technique in ZSL without requiring modifications in its pipeline (apart from hyper-parameters tuning). Also, a favorable

performance is scored by *VisEn* when directly comparing to state-of-the-art methods on AWA2 and CUB databases.

In summary, the contributions of this Chapter are threefold:

- We claim that visual embeddings trained for classification are sub-optimal in zero-shot learning. Instead, we use a captioner, predicting attributes at the image-level, to enhance visual embeddings and allow them to capture visually-grounded semantic cues;
- With respect to zero-shot captioning [72; 73; 74; 75; 76; 77; 78], we train our captioner in a weaker supervised regime which is compatible with ZSL benchmarks. Even when attributes are not annotated at the image-level, we take advantage of attributes labeled as incompatible with a given class to deduct the visual absence of the same attribute in any image of that class, generating instance-level supervision for free;
- Our enhanced visual embeddings, called *VisEn*, can replace default ones in a plug-and-play fashion, without requiring any change in the computational pipeline (apart from hyper-parameters tuning). Further, *VisEn* is capable of improving the performance of existing methods, overall scoring a favorable performance against state-of-the-art methods on AWA2 [80] and CUB [79] datasets.

3.2 Background and Related Work

In this Section, we will briefly refer to background material and related work to spot the factors of novelty of the proposed method with respect to the works existing in the literature.

Metric Learning for Zero-Shot Learning. All inductive zero-shot methods by metric learning can be framed as follows:

1. Pre-computed visual features $\mathbf{v}$ are used to encode input data;
2. Semantic embeddings $\mathbf{s}$ annotate the level of coherence in between a list of attributes and each seen/unseen class to recognize;
3. A metric function Φ is learnt. Usually called *compatibility function*, Φ is optimized in order to match $\mathbf{v}$ and $\mathbf{s}$ if they correspond to the same *seen* class.

At the inference stage, $\Phi(\tilde{\mathbf{v}}, \mathbf{s}^u_j)$ is computed on top of the test instance $\tilde{\mathbf{v}}$ and all semantic embeddings $\mathbf{s}^u_j$, where j indexes unseen classes. Hence, the class $j^\star$ is predicted if $\Phi(\tilde{\mathbf{v}}, \mathbf{s}^u_{j^\star})$

scores better than $\Phi(\tilde{\mathbf{v}}, \mathbf{s}^{\mathrm{u}}_j)$ for $j \neq j^\star$. In order to design the metric Φ for learning, different approaches have been attempted, by either considering bilinear functions [81; 82; 83; 84; 85], hidden embeddings models [61; 62; 63; 64], dictionary learning [65; 66; 67] and eventually shallow linear networks to project visual onto semantic embeddings [68; 69; 70; 71].

Differently to ZSL by metric learning in which visual embeddings are pre-trained for classification, here we pre-trained them by using a captioner to predict attributes. As a result, our enhanced visual embeddings are expected to be richer in semantic patterns, easing the metric learning stage without any change in its computational pipeline (apart from hyper-parameters tuning).

Zero-shot captioning. Natural language processing (NLP) has been shown to successfully generate human-like captions for object and categories never seen before. It was successfully applied to both images [72; 73; 74; 75; 76] and videos [77; 78]. The mainstream approaches leverage Recurrent Neural Networks with long-short memory units (LSTM), due to their remarkable effectiveness in NLP. The LSTM is usually fed with some intermediate representation learned from an encoder which processes raw data in an end-to-end fashion (for images, CNNs are usually adopted). Crucially, the common operative setup in zero-shot captioning is the annotation of each instance with many alternative extended descriptions (*i.e.*, up to 15 sentences provided by annotators) [72; 73; 74; 75; 76; 77; 78].

Differently, in this paper, we train a captioner in the weaker supervised regime available in ZSL benchmarks: when available, we utilize the instance-level guidance about the presence/absence of attributes within a fixed list. Even when attributes are not visually grounded (since annotated at the class level only), we are still capable of training our captioner despite this extremely weakly supervised regime. We do so by using the semantic incompatibility of an attribute and a class to get for free the visual absence of the same attributes in all instances of that class. In doing so, we can find an example of self-supervision [86; 87; 88]. In fact, we exploit an auxiliary task (here, captioning) which is solved with no need for additional supervision and is preparatory for the original task of interest (here, ZSL).

3.3 Weakly Supervised Captioner for ZSL

In this Section, we present how we trained a captioner to predict attributes at the instance level in the weakly supervised setup of ZSL benchmarks, relaxing the strict supervised regime which is commonly adopted in zero-shot captioning methods [72; 73; 74; 75; 76; 77; 78].

3.3.1 Datasets

Caltech-UCSD Birds 200 (CUB) [79]. CUB is a fine-grained dataset of 200 bird species, most of which are typical in North America. Coherently with the ZSL literature, we adopted the 2011 release in which 11788 images are available. For each, up to 5 Amazon Mechanical Turkers annotated a list of 312 attributes which provides an expert level characterization of each image, specifying minute details (such as colored spots on the neck/wings) which are fundamental to disambiguate between classes, *i.e.*, bird species.

Animals with Attributes 2 (AWA2) [80]. AWA2 is a coarse dataset composed of 50 different classes of animals (such as "polar bear", "zebra", "giraffe", "otter", etc. ...). To describe each class, a list of 85 attributes is provided by following Osherson's matrix (OM) [60] for class/attributes correspondence. Some values in the OM are not specified (and set to -1, like "black-colored" for the class "antelope"). The remaining entries of the OM contain a value scaled in the range [0,100], to rank how much an attribute is prototypical for that class (*e.g.*, the attribute "spots" for the class "dalmatian" has a value of 100 in the OM).

3.3.2 Implementation Details

As similarly done in [72; 73; 74; 75; 76], we employed an end-to-end trainable captioner which is composed of two modules: the first encodes image information in a feature vector, the second generates the caption. As the first module, here we used a ResNet-152 Convolutional Neural Network, pre-trained on the Imagenet ILSVRC2012: the 2048-dimensional fully connected (FC) layer - right before logits in ResNet-152 architecture - is linearly transformed into a 256-dimensional FC layer. This latter encoding is then used as the initial state for our uni-directional LSTM captioner (256-dimensional hidden state), which is our second module. A visualization of the adopted architecture is provided in Fig. 3.2.

Since CUB dataset provides attributes annotated at the image level, we directly used those annotations for training, by considering one attribute to be present if at least one Turker annotated it. For instance, "has_bill_shape::spatulate", "has_wing_color::brown", "has_wing_color::grey", "has_wing_color::buff", "has_upperparts_color::brown" and "has_upperparts_color::buff" are some of the ground truth attributes for the image `Black_Footed_Albatross_0089_796069.jpg` which is depicted in Fig. 3.2.

On the contrary, on AWA2, attributes are only annotated at the class level and, consequently, we do not have guidance about which attributes are actually present in which images. In fact, we are only given a confidence value for each attribute (like "quadrupedal") and each class (like "zebra), but this score is not capable of telling which images of the AWA2 dataset

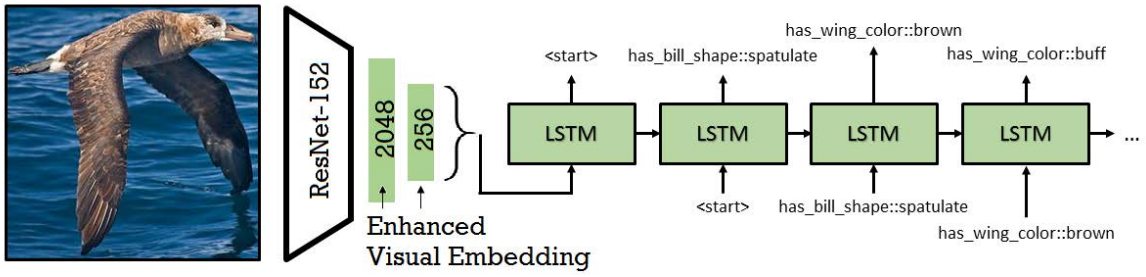


Figure 3.2 Architecture of our weakly supervised captioner.

depict a zebra with visible legs.

Differently, it is advantageous to consider attributes which are semantically incoherent with a certain class (such as "black-colored" for the class "polar bear"). Then, we can assume that those incoherent attributes will be visually absent in all instances of that specific class (since no realistic photo will depict a black polar bear). This consideration is crucial to cast semantic attributes provided at the class level into (absence of) visual attributes annotate at the image level. Attributes that are annotated as incoherent with respect to a certain class (on average, 28.6% of attributes per class on AWA2) are labeled as missing for each instance of that class. On the contrary, for the remaining attributes, since we can not draw a better conclusion from the available annotations, we assume them to be always present in the corresponding instances. This is the weak supervision that negatively relies on the absence of attributes only and that we can generate for free from AWA2 benchmark for the sake of training our captioner.

On both CUB and AWA2, consistently to the ZSL setup, training is done only on the images belonging to the seen classes, accordingly to the proposed splits of [89].

3.3.3 Attribute Prediction: Results

In this Section, we validate the performance of our weakly supervised captioner for attribute prediction, in both quantitative and qualitative terms.

Quantitative Results. We adopted the same binary classification framework of earlier ZSL models [60; 81]. That is, on CUB database, attributes by Turkers are used as ground truth and compared with instance-level predictions. The final reported performance is averaged across all 312 attributes and all training/testing images. Differently, on AWA2, we compare the binary predictions of our captioner (on top of a certain image I) with a binarization of the class-level attributes related to I. As in CUB, classification performance is averaged across the 85 attributes and all instances (in both training and testing).

Table 3.1 reports such classification accuracy values: our captioner is able to sharply improve in performance both [60] (+22.12% on CUB and +8.96% on AWA2) and [81] (+27.52% on CUB and +8.96% on AWA2). The sharper margin is registered on CUB dataset: on AWA2, in fact, we do not have a precise instance-level attributes supervision to train our captioner, whereas on CUB we do.

	Training Set	Test Set		
	Captioner	Captioner	[60]	[81]
CUB	*87.26%*	**86.92%**	64.8%	59.4%
AWA2	*98.98%*	**81.66%**	72.7%	72.7%

Table 3.1 Attribute prediction in CUB and AWA2 datasets. The performance of the captioner on the training set is in italic, we highlighted in bold the best testing accuracy in attribute prediction among the captioner and the ZSL paradigms [60] and [81].

Qualitative Results. To visualize the image features learned from the CNN module of our captioner, we take advantage of t-SNE [90].

t-SNE is the state-of-the-art technique to obtain a bi-dimensional visualization of arbitrary feature encodings. We run t-SNE on top of the 256-dimensional feature vector extracted from the CNN-module of our captioner trained on CUB database: the result is a set of bi-dimensional points $(x_i, y_i) \in \mathbb{R}^2$, each of which corresponding to the image I_i of CUB, $i = 1, \ldots, 11788$. Then, we used (x_i, y_i) as anchor points where to plot I_i: in this way, we can embed all CUB images into a planar representation such that two nearby images correspond to features that are close to each other in the visual space (according to t-SNE).

To do so, we quantized (x_i, y_i) into a grid of integers point $(r_i, c_i) \in \mathbb{Z}^2$, the latter being used to align the first pixel of I_i while plotting it. Images I_i have been spatially rescaled to 50×50, preserving the original RGB color space. In order to handle overlap between images, we operated a stretching along the c_i-th coordinate of all our anchors.

The results of this visualization are provided in Fig. 3.3 and 3.4, where we compare against the analogous procedure applied to the usual visual embeddings adopted for ZSL [89]: 2048-dimensional features extracted from a ResNet-101 model trained for classification over the seen classes.

ResNet-101 seems to encode similarly birds which have a similar shape, but different colors (and, therefore, different species - Fig. 3.4, orange box). In addition, when using the same descriptor, sometimes, birds appear to be clustered together accordingly to the sky in the background (Fig. 3.4, blue box).

Differently, using the 256-dimensional embedding of our captioner, we can better capture

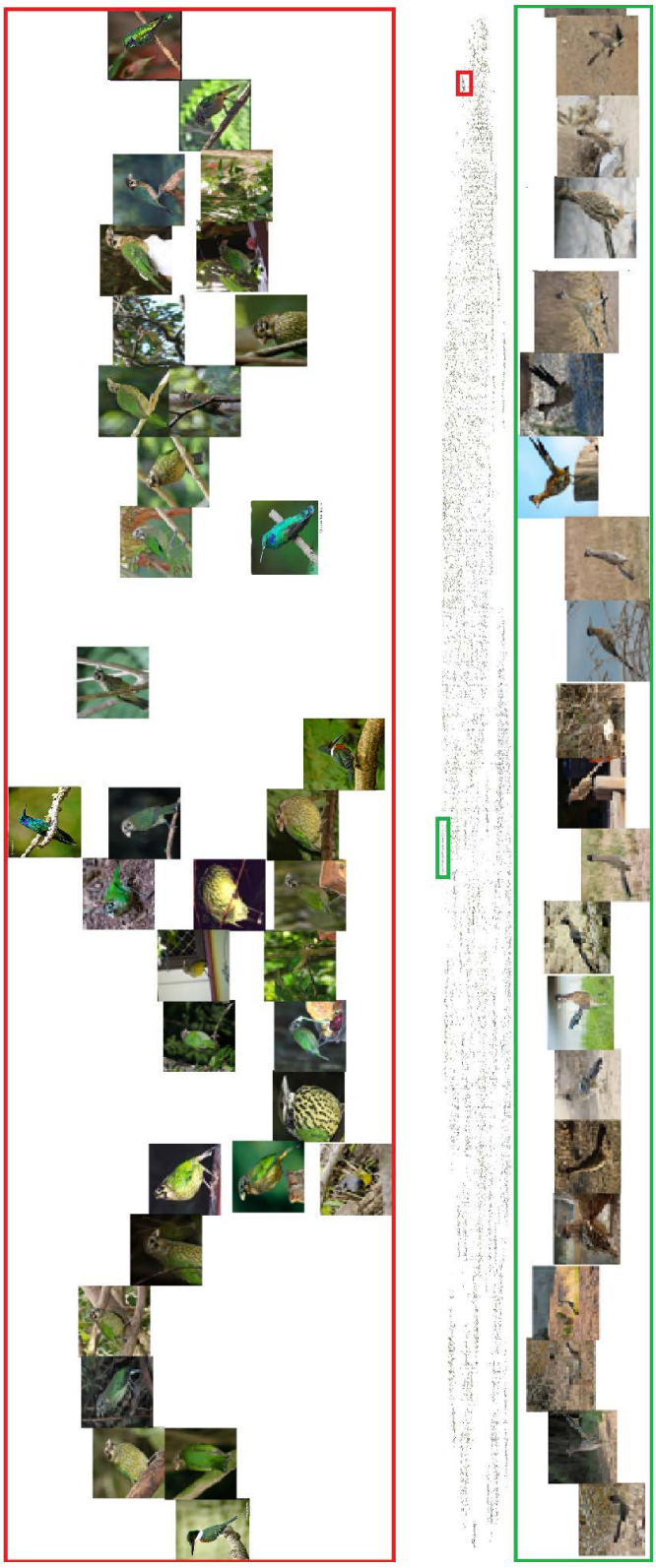


Figure 3.3 t-SNE visualizations of enhanced versus default semantic embedding for ZSL. Our enhanced visual embeddings finely learn the attribute "having-green-wings" (red box) and cluster birds with "long beack"+"long neck"+"long tail" in a color-consistent manner, so preserving species.

Figure 3.4 t-SNE visualizations of enhanced versus default semantic embedding for ZSL. Classical visual embeddings used in ZSL seems fooled by “contours”, mixing up different species which very different colors (orange box). Sometimes, the sky in the background compromises a correct embedding for species (blue box).

semantic patterns: in fact, we can observe a nice clustering effect of birds with green wings (Fig. 3.3, red box). Also, our enhanced visual embeddings seem to cluster birds with similar shape (long tail and neck, elongated body) *accordingly* to their colorization and the respective class as well (Fig. 3.3, green box).

In summary, even if using a visual embedding that is one order of magnitude smaller than a baseline one (ResNet-101), we are capable of capturing more semantic patterns.

3.4 Enhancing Visual Embeddings: Benchmarking the State-of-the-Art in ZSL

In this Section, we provide quantitative results to assess the effectiveness of our proposed enhancement of visual embeddings (*VisEn*). Precisely, on AWA2 and CUB benchmarks, we prove that three popular approaches in zero-shot learning [64; 66; 70] are sharply boosted in performance when replacing classical visual embeddings with ours. Apart from hyper-parameters tuning, such replacement does not require changes in the computational pipeline. Further, in Section 3.4.2, we set up a broad comparison between *VisEn* and the state-of-the-art performance in ZSL.

3.4.1 Ablation study

For our ablation, we considered the following approaches for ZSL:

• *Synthesized Classifiers* (SynC) [64] learns a latent embedding in which to combine visual and semantic embeddings in a max margin sense, by means of three different hinge losses: one-versus-one (OVO), Crammer and Singer (CS) [91] and a structured output SVM loss (struct).

• *Semantic Auto-Encoder* (SAE) [70] proposes a shallow linear encoder-decoder network to project visual embeddings into semantic ones (through a trainable projection matrix $\mathbf{W}$) and then reconstruct the visual embeddings from the semantic ones (through $\mathbf{W}^\top$). By using either the projection learned from the encoder or the decoder, the compatibility function is the Frobenius norm between ground truth and predictions.

• *Coupled Dictionary Learning* (CDL) [66] learns a latent embedding in which semantic embeddings are projected and, by means of a synchronous dictionary learning pipeline, visual embeddings are mapped onto the latent ones and vice-versa. All such projections can be combined (or even separately used) at the inference stage which is configured as an Euclidean nearest neighbors search.

Dataset: AWA2				
		SVM loss		
Visual Embedding	*d*	OVO	CS	struct
GoogleNet	1024	52.6%	53.4%	**59.0%**
ResNet-101	2048	53.0%	53.7%	**59.0%**
VisEn (ours)	*256*	*50.7%*	*51.0%*	*52.1%*
VisEn (ours)	*2048*	***54.6%***	***54.4%***	***59.0%***
Dataset: CUB				
		SVM loss		
Visual Embedding	*d*	OVO	CS	struct
GoogleNet	1024	53.4%	51.6%	54.5%
ResNet-101	2048	55.6%	49.0%	53.9%
VisEn (ours)	*256*	***59.4%***	***53.2%***	***54.6%***

Table 3.2 **Comparison with SynC** [64]. The performance of our visual enhancement (*VisEn*) is in italic, the best performance is in bold. In this table, we used the proposed split (PS) by [89].

For SynC, SAE and CDL, we optimized from scratches their compatibility functions by using publicly available code [2]. We used default semantic embeddings (specifically, the ones provided in [89])

In addition, we ablate on several factors: the OVO, CS and struct losses for SynC, all the possible combinations of projections learned by CDL and the alternative usage of the encoder or the decoder for SAE. Moreover, for the latter method, since it is a crucial aspect in ZSL [89], we tried different manners of splitting seen and unseen classes: a random extraction of 40 classes for training and 10 for testing (as commonly done in literature) in addition to standard (SS) and proposed splits (PS) from [89].

Results are reported in Table 3.2 (for Sync), Table 3.3 (for SAE) and Table 3.4 (for CDL). In all of them, *d* denotes the size of the adopted visual embedding.

Discussion

In Table 3.2, *VisEn* exactly matches the performance of GoogleNet and ResNet-101 features on AWA2 with the OVO loss, while in all other cases is superior to both descriptors: the mean average improvement is 1.7% and 1.8% over them, respectively. In Table 3.3, *VisEn* improves SAE on the PS for AWA2 and in all splits for CUB. Finally, with respect to CDL,

[2] SynC: `https://github.com/pujols/zero-shot-learning`; SAE: `https://github.com/Elyorcv/SAE`; CDL: `http://vipl.ict.ac.cn/resources/codes/code/ECCV2018_CDL_code_release.rar`

	Visual Embedding	d	Dataset: AWA2 40/10	AWA2 SS [89]	AWA2 PS [89]	Dataset: CUB 40/10	CUB SS [89]	CUB PS [89]
Encoder	GoogleNet	1024	**84.7%**	78.5%	63.5%	61.4%	44.4%	46.2%
	ResNet-101	2048	79.6%	80.0%	64.0%	57.0%	54.4%	57.9%
	VisEn (ours)	*256*	*77.3%*	*77.3%*	*57.3%*	***62.5%***	***65.4%***	***58.6%***
	VisEn (ours)	*2048*	*82.9%*	***80.3%***	***65.7%***	-	-	-
Decoder	GoogleNet	1024	**84.0%**	**80.1%**	63.1%	60.9%	44.2%	46.2%
	ResNet-101	2048	79.0%	79.0%	63.4%	57.5%	54.8%	58.6%
	VisEn (ours)	*256*	*77.4%*	*77.4%*	*57.8%*	***63.2%***	***66.2%***	***59.2%***
	VisEn (ours)	*2048*	*80.3%*	*78.3%*	***64.1%***	-	-	-

Table 3.3 **Comparison with SAE** [70]. The performance of our visual enhancement (*VisEn*) is in italic, the best performance is in bold. In this case, we report the separate performance of the encoder and the decoder. Also, we ablate on several splits of seen and unseen classes: we consider the standard (SS) and proposed splits (PS) provided by [89] and the same 40/10 split (10 random classes as unseen ones, the remaining as seen ones) used in [70].

	Dataset: AWA2 ResNet-101 d = 2048	AWA2 *VisEn (ours)* d = 2048	AWA2 *VisEn (ours)* d = 256	Dataset: CUB ResNet-101 d = 2048	CUB *VisEn (ours)* d = 256
v	**63.8%** ± 4.3%	*59.6% ± 6.8%*	*62.0% ± 2.9%*	40.0% ± 2.3%	***57.6%*** *± 1.5%*
a	61.4% ± 1.9%	*62.6% ± 2.1%*	***63.1%*** *± 1.2%*	50.2% ± 1.8%	***51.5%*** *± 0.8%*
l	53.9% ± 2.7%	*51.4% ± 3.2%*	***57.4%*** *± 1.0%*	40.3% ± 2.2%	***50.5%*** *± 2.7%*
v + a	**66.8%** ± 2.7%	*65.3% ± 2.9%*	*63.9% ± 1.6%*	54.6% ± 1.9%	***58.3%*** *± 0.9%*
a+l	59.1% ± 1.7%	*55.6% ± 2.4%*	***61.4%*** *± 0.7%*	46.3% ± 2.0%	***51.9%*** *± 1.2%*
v+l	62.5% ± 1.7%	***66.5%*** *± 3.0%*	*62.0% ± 0.7%*	49.5% ± 2.0%	***57.1%*** *± 1.1%*
v+a+l	62.6% ± 1.5%	*59.5% ± 2.4%*	***62.7%*** *± 0.8%*	50.7% ± 1.9%	***56.5%*** *± 0.9%*

Table 3.4 **Comparison with CDL** [66]. The performance of our visual enhancement (*VisEn*) is in italic, the best performance is in bold. In this table, we adopted the proposed splits (PS) by [89] and compare different visual embeddings (whose dimensionality d is reported beneath for completeness). Also, we ablate on which combination of the three projection (visual embedding v, attributes a or latent embedding l) is used for the nearest neighbor search during inference. Since CDL leverages an iterated optimization, we provide mean and standard deviation of testing accuracy across iterations (whose number was fixed to 50 for AWA2 and 100 for CUB, as in [66]).

while accounting for all different projections setup (the different rows in Table 3.4), the 256-dimensional *VisEn* improves ResNet-101 features in 6 cases out of 7 (AWA2) and in 7 cases out of 7 on CUB.

More in details:

- On the AWA2 benchmark, despite the captioner was trained by only using negative supervision about the absence of attributes, *VisEn* was frequently able to match the performance of descriptors which trained (for classification) with full supervision.

    1. When using SAE Encoder and Decoder (with 40/10 split), GoogleNet features are +1.8% and +2.7 better than 2048-dim *VisEn*, respectively. In the very same setup, 2048-dim *VisEn* improves ResNet-101 features by +1.3% and by +3.3% on the PS splits, when using the SAE Decoder and SAE Encoder, respectively;
    2. Often, the 2048-dimensional *VisEn* are slightly superior to both GoogleNet and ResNet (SynC-OVO, SynC-CS, SAE Encoder SS, SAE Encoer PS, SAE Decoder PS, and CDL settings) guaranteeing an improvement of about one/two percentage points;
    3. For both SynC and SAE, 256-dimensional *VisEn* seems sub-optimal but, interestingly, the very same descriptor is able to score a remarkable performance in conjunction with CDL: despite one order of magnitude less, it is capable of improving ResNet-101 and GoogleNet features on the a, l, a+l and v+a+l settings, even by 3.5%;
    4. We can observe that for both Sync and SAE, 2048-dimensional *VisEn* are always better than 256-dimensional ones, whereas, for CDL, the opposite trend is registered. This seems to suggest that CDL (which adopts dictionary learning) is capable of optimally perform even when fed with a relatively low-dimensional visual embedding. Differently, by either performing a latent embedding (Sync) or a direct mapping in between visual and semantic embeddings (SAE), a bigger dimensional visual embedding is required.

- On CUB, the performance is undoubtedly coherent in its trend: 256-dim *VisEn* is *always* superior to GoogleNet and ResNet-101 features.

    1. In Table 3.2, while averaging across OVO, CS and struct losses, the average improvement of *VisEn* is +2.9% with respect to ResNet-101 features and +2.6% with respect to GoogleNet ones;

	Dataset: AWA2		Dataset: CUB	
	SS	PS	SS	PS
DAP [60]	58.7	46.1	37.5	40.0
IAP [60]	46.9	35.9	27.1	24.0
CMT [69]	66.3	37.9	37.3	34.6
DEVISE [82]	68.6	59.7	53.2	52.0
ConSE [68]	67.8	44.5	36.7	34.3
SSE [61]	67.5	61.0	43.7	43.9
SJE [83]	69.5	61.9	55.3	53.9
ALE [81]	<u>80.3</u>	62.5	53.2	54.9
ESZSL [85]	75.6	58.6	55.1	53.9
LatEM [84]	68.7	55.8	49.4	49.3
SynC [64]	75.4	59.7	53.0	54.6
SAE [70]	**80.7**	54.1	33.4	33.3
expZSL [62]	79.3	63.8	53.0	49.3
LDA [65]	-	56.6	-	-
CDL [66]	-	**69.9**	-	54.5
VdSA [71]	-	-	<u>56.7</u>	-
PSR [63]	-	63.8	-	<u>56.8</u>
VisEn (ours)	*<u>80.3</u>*	*<u>65.7</u>*	***65.4***	***58.6***

Table 3.5 **Comparison with the state-of-the-art in inductive ZSL by metric learning**. The first and the second best accuracy values are highlighted in bold and underlined, respectively.

2. In Table 3.3, across the 40/10, SS and PS splits and the usage of the Encoder or Decoder, SAE is improved by +12.0% with respect to GoogleNet features and by +5.9% with respect to ResNet-101 features;
3. Finally, in Table 3.4, across all combinations of projections with CDL, ResNet-101 features are improved by +7.4% on average

All such systematic improvements on CUB are even more impressive if considering that they were achieved with a visual embedding of one order of magnitude less than baseline ones.

3.4.2 Comparison with the State-of-the-Art in Inductive ZSL by Metric Learning

In this Section, we directly compare the proposed enhancement of visual embeddings (*VisEn*) with state-of-the-art approaches in inductive zero-shot learning via metric learning.

Precisely, we compare *VisEn* - fed into a SAE encoder [70] - with direct and indirect attributes' prediction (DAP and IAP) [60], the cross-modal transfer (CMT) [69], the hybrid semantic and visual embedding (DEVISE) proposed in [82], the convex combination of semantic classifiers (ConSE) [68], the semantic similarity-preserving embedding (SSE) of [61], the Structured Joint Embedding (SJE)[83], label embedding for zero-shot image classification (ALE) [81], the regularized least square method for ZSL (ESZSL) [85], the latent embedding model which solves ZSL through ranking (LATEM) [84], the synthesized classifiers learnt in a max margin sense (SynC) [64]. Among the most recent ones, we considered the shallow semantic autoencoder (SAE) [70], the approach of ZSL which uses exponential family distributions (expZSL) [62], the discriminative learning of latent attributes (LDA) [65], the coupled dictionary learning approach (CDL) [66], the visually-driven semantic augmentation [71] and the metric learning approach which preserves semantic relations (PSR) [63].

Results of this extended comparison are reported in Table 3.5. For AWA2 and CUB, we utilized the standard (SS) and proposed splits (PS) of [89]. By doing so, we can leverage the work of the survey [58] which reported the performance of DAP, IAP, CMT, DEVISE, ConSE, SSE, SJE, ALE, ESZSL, LatEM, Sync, SAE and expZSL for those splits while using ResNet-101 features as visual embeddings. Also for LDA, CDL, VdSA and PSR, we are reporting published classification accuracy values extracted from the respective publications.

Discussion

Overall, our proposed approach scores a solid performance, so that *VisEn* locates itself as the second-best scoring method on AWA2 and the best one on CUB.

On AWA2, *VisEn* sets a second-best performance on both SS and PS. The performance is still notable due to the following aspect. Existing state-of-the-art methods extract their visual embeddings in a fully supervised regime, whereas, differently, our captioner was forced to operate in a more challenging setting in which only negative weak supervision was provided (absence of attributes). Despite this restriction made the comparison more demanding for us, still, our proposed enhancement of visual embedding scores a favorable performance with respect to the state-of-the-art.

Instead, on CUB, due to the availability of visually grounded attributes, *VisEn* can express its full potential and proves the effectiveness of doing metric learning by combining the coherence score of an attribute for a class (semantic embedding) and the visually-grounded predicted presence of an attribute inside an image of that class (*VisEn*). In fact, on CUB, our

proposed enhancement registers an improvement over the previous best scoring method of +2.2% (on PS) and +9.7% (on SS).

3.5 Conclusions

In this Chapter, we propose to replace the usual visual embeddings in ZSL (which are trained for fully supervised classification) with the intermediate representation of an end-to-end captioner that predicts attributes at the instance-level. Without the usage of the usual supervision adopted in zero-shot captioning (multiple extended descriptions per instance), we still proved the effectiveness of training a captioner in the weakly supervised regime which is typical of zero-shot recognition (list of attributes). In fact, even when attributes are annotated at the class level, we can rely on the semantic incompatibility between a given attribute and a certain class: all instances of that class do not contain the specific attribute. Leveraging this observation, we generate for free visually-grounded supervision to train our captioner, using it to extract visual embeddings that are richer in semantic content with respect to baseline ones (ResNet-101 features). Our proposed enhancement of visual embeddings *VisEn* is compatible with any generic ZSL method, without requiring changes in its pipeline (apart from hyper-parameter tuning). We proved that *VisEn* systematically improves the recognition performance of three popular approaches in ZSL [64; 66; 70], eventually outperforming classical GoogleNet and ResNet-101 features. Experimentally, *VisEn* achieves the second-best performance on AWA2 benchmark, despite *VisEn* were obtained by means of negative weak supervision. Differently, on CUB, leveraging clean instance-level annotations, we sharply boosted the best scoring methods on CUB by +2.2% and +9.7%, on both standard/proposed splits of [89].

3.6 Future Works

The activities discussed in this Chapter are subject of active research and many of the identified issues need further investigation and experimentation. Zero Shot Learning field itself has a big room for improvements since a huge gap still exists before human-level performances can be reached. Some of the most evident difficulties to tackle are the following:

- **Visual Embeddings and Semantic Embeddings are not balanced**

  In most of the datasets used by the Zero Shot Learning community, attributes are annotated at the class level only and instance level annotation is not available. This

means that, during training, a single class semantic feature $\mathbf{C_{SF}}$ has to be put in relation with hundreds of visual features $\mathbf{I_{VF}}$ representing the input image, and vice-versa. What kind of relationship should we expect to establish between these two domains?

- **Semantic Embeddings are not always visually grounded**

  In many datasets used by Zero Shot Learning community, attributes are annotated by providing a measure of coherence between each attribute and each class. This makes semantic embedding not visually grounded: e.g. we can expect a strong semantic coherence between the attribute "quadrupedal" and the class "zebra", but such information can fool a Zero Shot Learning model when recognizing an image of a zebra only depicting its upper body, whose legs are not visible. Is it possible to develop a system that takes into account this fundamental aspect of Zero Shot Learning, to better exploit the annotated dataset available and the realistic scenarios?

- **Feature Learning**

  Many state of the art Zero Shot Learning methods extract $\mathbf{I_{VF}}$ from a CNN trained for classification, without investigating the possibility that Zero Shot Learning pipeline can actually benefit from visual features generated differently. Similarly, it is possible that also the semantic embedding can be improved knowing that they will be used for a visual-recognition task. $\mathbf{I_{VF}}$ is visually grounded by construction, but we have no grants about the strength of its semantic content. Similarly, $\mathbf{C_{SF}}$ has a rich semantic content by construction, but it might not be visually grounded.

  The introduction of an intermediate feature representation $\mathbf{I_{SF}}$ (Fig. 3.5) might be used as a bridge between visual and semantic content, helping in transferring information between the two domains [64; 71; 92], by selecting most relevant contents in $\mathbf{I_{VF}}$ and $\mathbf{C_{SF}}$, or in adding interpretability in the Zero Shot Learning architecture [93; 94].

  A better understanding of the visual+semantic space, where the Zero Shot Learning problem is defined, could finally bring to a better feature design approach.

As discussed in [95], we argued that extracting visual features for Zero Shot Learning through captioning is a more effective way of building them, compared to the mainstream approach of using visual features designed for image classification. The aforementioned results are very encouraging, nevertheless, we still need a deeper analysis of the following aspects:

- Is there a metric we can use for judging the quality of the semantic content of the obtained visual features $\mathbf{I_{VF}}$? Can image saliency detection methods and attention

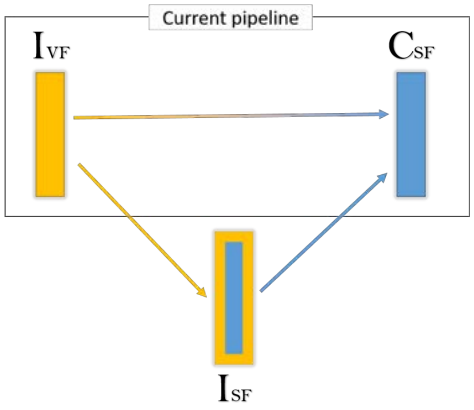


Figure 3.5 The introduction of an intermediate feature representation $\mathbf{I_{SF}}$ might be used as a bridge between visual and semantic content, leading to better recognition in the zero-shot setup.

models give support during $\mathbf{I_{VF}}$ training and in verifying the visually-grounded property of the class semantic feature $\mathbf{C_{SF}}$?

- Can the proposed approach benefit of a Bayesian formulation, for explicitly modeling the (eventual) lack of visual grounding in semantic embedding?
- By explicitly predicting attributes from each image, is there the opportunity to use a joint approach for Zero Shot Learning and Attribute Prediction? In such a case, we could investigate which of the two tasks is more effective in helping the other.

Trying to find an answer to these questions and to tackle the listed difficulties is an ongoing research activity.

Part II: Applying State-of-the-Art AI for Social Good

In this Section, I will describe some of the applications of state-of-the-art Deep Learning and Computer Vision models I have developed during my PhD.

As discussed in the Introduction, the technologies currently available can address and tackle many real-world problems, proposing innovative solutions or improving the already existing methods.

In Chapter 4, I will discuss about the analysis of face-touching behavior. I will describe the annotated dataset that we publicly distributed and several computer vision methods that we implemented and made available to the scientific community in order to set a baseline performance and promote additional research activities in this relevant topic.

In Chapter 5, I will describe the privacy-preserving solution we developed for estimating the "Social Distance" and its violations, given a single uncalibrated image in unconstrained scenarios.

In Chapter 6, I will present a Computer Vision methodology developed in collaboration with the Egyptian Museum of Turin for digitally bandage unwrapping mummies. Starting from their CT scans, we were able to virtually remove bandages from the mummies' images, supporting the archaeologists during their analysis and avoiding the devastating and irreversible process of physically unwrapping the bandages for removing amulets and jewels from the body.

Chapter 4

AI and Covid 19: Analysis of Face-Touching Behavior

4.1 Introduction

People touch their face frequently, even without realizing, while wiping their eyes, scratching their noses/face, biting their nails, twirling their mustaches or in the presence of even mild abrasions on the face [96; 97]. Hands are often kept close to the face during conversations, and face touch might occur for several reasons, such as embarrassment, stress, anxiety, fatigue, surprise and so forth [98; 99; 100]. This behavior is very persistent in our daily life as studies based on manual observation reported that people touch their face between nine to 54 times per hour, on average [96; 97; 101; 102].

This aspect is of clear relevance in the current global pandemic emergency as contaminated hands are a potential carrier for the dissemination of respiratory diseases, e.g., influenza and coronavirus [96; 97]. During the influenza A (H1N1) pandemic, face-touching behavior in the community was commonly detected. According to [103], individuals were touching their faces on average 3.3 times per hour. Similarly, during the Covid-19 pandemic, avoiding face-touching behavior has become one of the most essential preventive actions that a person can do [104]. A very recent multi-disciplinary survey paper [104] covering more than 100 papers (e.g., in PubMed, Scopus, PsychInfo, EconLit, Science Direct and Google Scholar) about hand-washing, face-touching, self-isolation, public-spirited behavior, and responses to crisis communication highlights that there are effective behavioral interventions to increase hand-washing, but none for reducing face-touching. The conclusion of the survey [104] regarding face-touching behavior is that some principles of behavior change models might be applied to reduce it. However, changing a behavior includes altering physical and

social environments as well as a person's mindset [105; 106]. In other words, education and information might be not enough on their own.

Machine learning and computer vision domains offer various methods that are able to solve very complex problems by processing noisy, incomplete, small/big scale data in a short time and sometimes achieve better performance than human experts. Particularly, human detection and pose estimation from images can be helpful to detect face-touching behavior at distance and in a non-invasive manner. Additionally, the existing frameworks such as intention detection [107], trajectory forecasting [108] and early recognition of actions [109; 110] can be adapted to prevent face-touching behavior before it occurs. Differently, detection of face-touching behavior can be integrated into the recognition of affective states as well as social interactions, as such gestures are an important channel of nonverbal communication.

Nevertheless, the very first step for realizing all these aforementioned applications is the data collection. Existing datasets (e.g., [111]) do not include publicly available annotations for face-touching behavior. Collecting such annotations in real-world situations including a large variety of face-touching behavior is beneficial to assess the complexity of the task and to foster future research on this problem of recent relevance.

The main contribution of this Chapter [1] is introducing face-touching behavior annotations, precisely the behavior of a person touching his/her own face, which were collected for a dataset composed of small group social interactions in meeting environments [112]. The binary annotations (*face-touch* or *no-face-touch*) are for 64 videos (64 participants), each lasting between 12 to 30 minutes. In total, 74K and 2M video frames were labelled as *face-touch* and *no-face-touch*, respectively. Some frames having the face-touch label are given in Fig. 4.1. Annotations were performed by a total of 16 participants and the inter-rater agreement calculated for each video shows almost perfect agreement on average. The details of the dataset [112] and the annotations are given in Section 4.3.

Using the annotations collected, we tested several approaches to detect face-touching behavior in images. These methods are categorized into three, with the latest providing the best performance:

1. A rule-based approach;

2. Supervised learning and inference with hand-crafted features;

[1] This Chapter has been published as: "*Analysis of Face-Touching Behavior in Large Scale Social Interaction Dataset*", ACM International Conference on Multimodal Interaction (ICMI 2020) by C. Beyan, M. Bustreo, M. Shahid, G. L. Bailo, N. Carissimi, A. Del Bue.

Figure 4.1 Face-touching behavior examples.

3. Feature learning and inference with a deep neural network

The contributions of this study can be summarized as follows.

- We present face-touching behavior annotations for a dataset having natural small group interactions [112]. These annotations as well as the modalities extracted/used in the performed experimental analyses (e.g., pose estimation results, face and hand image locations, corresponding images, data splits) are made publicly available.
- This is the only publicly available dataset serving the largest amount of face-touching behavior annotations and containing the data belonging to the highest amount of participants so far included in the same tasks.
- We benchmark the collected annotations for the detection of face-touching behavior in images. Several methods were tested and their performances, which can be used as the baseline results by future studies, were compared.

The rest of this Chapter is organized as follows. The behavioral observation studies regarding face-touching behavior as well as the social/affective computing studies regarding recognition of facial occlusions including hand over face gesture analysis are reviewed in Section 4.2. The details of the dataset [112] and the annotations collected for face-touching behavior are explained in Section 4.3. Section 4.4 describes the experimental setting and the approaches that were tested on the dataset [112] and the corresponding face-touch annotations. Section 4.5 includes the results of the methods applied. Finally, in Section 4.6 we conclude the Chapter with a summary and a discussion including the future directions that can be implemented given the annotations introduced.

4.2 Related work

An observation study regarding face-touching behavior was undertaken in [96]. The videotape recordings of the medical students were manually analyzed, concluding that on average, each of the 26 observed students touched their faces 23 times per hour. Similarly, in [97], 10 subjects were recorded for three hours while performing office-type work in isolation from other people. The number of contacts to the eyes, nostrils, and lips was counted manually, showing that the average total contact rate per hour was 15.7. As a more recent study, Morita et al. [113] investigated the relationship between face-touching behavior frequency in a simulated cabin of a train that they had built. This experiment was conducted 12 times, each lasting 30 minutes. In total 40 students were participated as the subjects whose face-touching behavior was observed. They found that the average face-touching frequency was 17.8 times per hour. Out of all types of face touches, mucosal contact (eyes, nose, mouth) was 42.2%. Moreover, the face-touching frequency was significantly higher for males than for females. For the latter, face-touching was significantly higher for the ones who did not wear makeup as compared to those who did.

These studies [96; 97; 113] show that face-touching behavior is frequent. Still, it is important to keep in mind that their conclusions are all based on manual inspection, which limits the number of subjects involved as well as the size of the data that can be analyzed. There is no published work specifically dedicated to automatic face-touching behavior analysis. However, some works on automatic recognition of facial occlusions, which are discussed in the following Section, involve face-touching behavior.

4.2.1 Automatic recognition of facial occlusions

Although facial occlusions are often treated as noise (e.g., discarded in recognition of face and affective states and removed from the detection of facial landmarks), there are still some works building the recognition systems considering the existence of facial occlusions [99; 100; 114; 115; 116]. For instance, Burgos-Artizzu et al. [114] detect facial landmarks from occluded faces and include the information extracted from them in their landmark location prediction method. Wu and Ji [115] use occlusion patterns as a constraint to predict facial landmark occlusions and landmark locations. Facial occlusions are estimated using a convolutional neural network trained on synthetic data by Saito et al. [116]. These methods [114; 115; 116] do not differentiate the types of facial occlusions such as occurring due to hands or due to objects.

Out of many possible facial occlusions, recognition of hand over face occlusions, which include face-touching behavior, has been considered by fewer studies [98; 99; 100]. The main reason for not gaining much attention is the lack of annotated natural datasets, since obtaining them is time demanding and expensive [100]. However, hand over face occlusions can provide important information such as for the recognition of some affective states: curiosity, frustration, boredom and etc., [98; 99; 100].

Cam3D dataset [111] contains video frames having hand over face occlusions as utilized by [98; 99] to test an automatic hand over face occlusion detection method. The considered occlusions include face-touching gestures as well. The dataset [111] was originally designed to elicit natural expressions. It is a 3D multi-modal corpus of natural complex mental states and includes labeled videos of spontaneous facial expressions and hand gestures of nine participants. It contains segmented videos, each showing a single event, e.g., a change in facial expression, head and body posture movement or hand gesture. The mean duration of video segments is six seconds. Unfortunately, hand over face gesture annotations are not publicly available. Furthermore, the supplied video segments are very short and the variety of the face occlusions, as well as the face-touching gestures, are limited as compared to our dataset [112]. It was mentioned in [98] that hand over face occlusion appears in 21% of the video segments. The hand covers the upper face and the lower face regions in 13% and 89% of the video segments, respectively, while both regions are covered in some videos. These results show that in natural interaction environment, hand over face occlusions are common and hands usually cover lower face regions, especially the chin, the mouth and lower cheeks, more than upper face regions.

Nojavanasghari et al. [100] synthesize naturalistic facial occlusions starting from an initial dataset of non-occluded faces and separate images of hands. The created dataset [100] includes 9.912 occluded faces such that 5.172 of them have hand over face occlusions (including face-touching behavior) and 4.740 images include other face occlusions, e.g., hair, scarf, glasses. Several methods: a kernel SVM, logistic regression and a deep neural network were tested to differentiate between hand over face occlusions and other types of occlusions.

More recently, Behera et al. [117] study the automatic detection of learner's nonverbal behaviors. The investigated behaviors include hand over face gestures as well. They used a dataset involving nine students either reading or problem-solving. The dataset has a total of 40 minutes of recordings and includes 320K images. Only 99K and 60K images were used for training and testing (no cross-validation) of the hand over face gesture detectors, respectively. Unlike our dataset [112], their dataset [117] does not include social interactions between participants and it is not publicly available. It is also limited as compared to our dataset [112]

in terms of the number of participants involved, the number of images collected, as well as the annotation procedure applied. Hand over face gestures were labeled by two annotators in total, while the criterion is whether participants' hand appear over the face or not. Each image was labeled by only one person and the reliability analysis of the annotations was not included. It is mentioned that the occurrence of hand over face gestures is on average 21.35% of the 40 minutes session and these gestures appear more frequently during problem-solving as compared to reading [117]. Behera et al. [117] compared several features: histogram of oriented gradients, features extracted from VGG-16, Inception-V3 and Inception ResNet-V2 when the classifier was SVM or a neural network. Inception-V3 features classified with the neural network performed better than the other methods considered in that study.

4.3 Dataset description

The new annotation collected regarding face-touching behavior is performed on the dataset presented in Beyan et al. [112]. The dataset [112] has been used for analyzing small group interactions (emergent leadership, leadership style [112; 118]) in terms of many nonverbal cues (visual activity, pose, gaze, speaking activity, prosody, etc.). As it contains natural discussions with no role-playing, the resulting gestures are completely natural (i.e., not instructed). Therefore, we expect that a detector trained on our dataset [112] would be more effective when it is tested on other real-world situations, especially compared to a detector trained on role-play face-touching gestures.

The face-touching annotations were performed for 64 videos (i.e., for all videos of the dataset [112]). Each video (1280×1024 pixels resolution with a frame rate of 20 frames per second (fps)) contains a single person, who was captured by a standard ethernet Axis camera located in front of her/him. The longest video lasts 30 minutes while the shortest video lasts 12 minutes. The total duration of the annotations is 393 minutes. There are 44 female and 20 male subjects in total.

4.3.1 Face-touching behavior annotations

The face-touching behavior annotations were performed using Computer Vision Annotation Tool (CVAT) [119; 120]. This is a free, open-source, web-based image and video annotation tool, which can be used for labeling data for computer vision tasks, e.g., object detection, image classification, and image segmentation. We customized its image tagging facility to

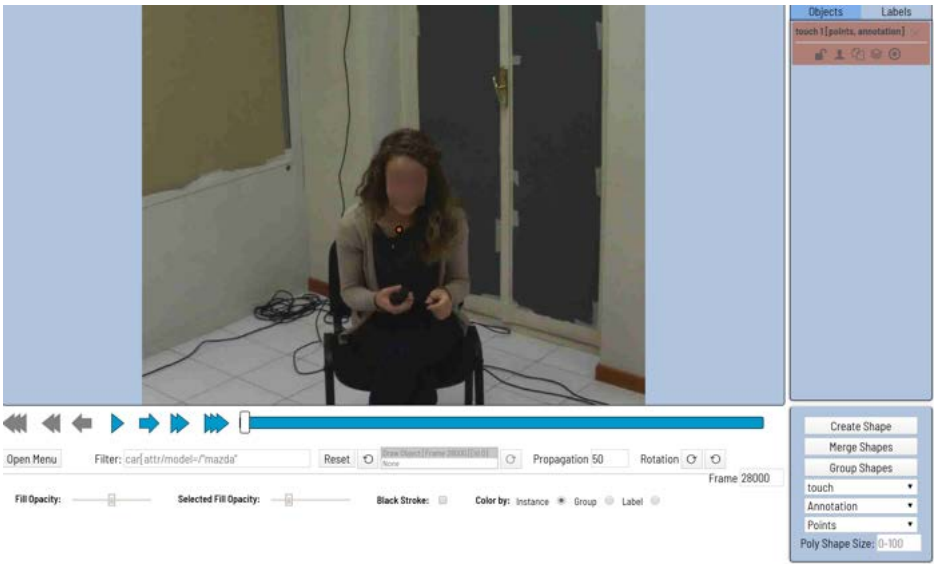


Figure 4.2 The graphical user interface that was used for the face-touching behavior annotation.

determine the video frames in which a subject is touching her/his face. In Fig. 4.2, the user interface used to annotate the face-touching behavior is given.

In total, 16 people participated in the annotation task, which was realized in two cycles. The face-touching behavior was described to the annotators as follows: touching any parts of the face (e.g., eyes, nose, mouth) including all the way up to the hair, and down under the chin by one or both of the hands is a face-touch, while touching hair, ears, behind the ears, and the neck are not included to face-touch. If a person is touching only her/his glasses without touching a part of his/her face, we asked annotators not to consider that as a face-touch.

In the first cycle, 10 participants were involved, each of them labeled 6 to 12 videos. Each video was labeled by one participant such that she/he observed all the frames of a video and determined the ones having face-touch (implying that the rest of the frames do not have face-touch). In CVAT, we defined one annotation task for one video, which was divided into small jobs having various numbers of frames, e.g., 100, 200, 500 or 1.000. In the second cycle, in total six participants (different from the ones involved in the first cycle) observed the jobs having at least one frame annotated as face-touch, without knowing the designated labels in the first cycle of the annotation. In other words, these six participants observed a part of each video but not all frames.

In total 2.081.232 (2M) frames were annotated in the first cycle. 516.100 (500K) of them were included in the second cycle of the annotations. In that way, each of the 516.100 frames

Table 4.1 Analysis of annotations. Full agreement means all three annotators agree with each other, majority agreement means two out of three annotators agree with each other for face-touch and not-face-touch classes. Values given with percentage (%) refers to the agreement in a single class.

	Full Agreement	Majority Agreement
Face-touch	70.459/74.367 (94%)	3.908/74.367 (6%)
No-face-touch	432.312/441.733 (98%)	9.421/441.733 (2%)

was annotated by three participants. For these frames, the final labels were determined by applying majority voting.

The analysis regarding the annotations performed by three participants are given in Table 4.1 in terms of two labels: face-touch and no-face-touch when there is a full agreement (i.e., all three annotators agree with each other) or majority agreement (i.e., two out of three annotators agree with each other).

As seen from Table 4.1, a significant number of frames has a full agreement. Given that the three participants who annotated the selected frames of a video are fixed, it is possible to calculate a measure of the agreement between them. For each video, we calculated Cohen's Kappa coefficient between each pair of participants and then took the average of them. For the total 64 videos, Cohen's Kappa scores are in the interval of [0.33-0.94]. The average and standard deviation of the Cohen's Kappa scores are 0.89 and 0.12, respectively. This means that the annotations per video change from fair agreement to almost perfect agreement [121].

The final annotations are composed of 2.006.865 (2M) frames labeled as no-face-touch and 74.367 (74K) frames labeled as face-touch. This results in a highly imbalanced dataset with the imbalanced ratio (calculated as the total number of face-touch frames divided by the total number of no-face-touch frames) equal to 0.04. The average number of frames labeled as face-touch is 1.162 frames per video. Two of the 64 videos do not have any frame labeled as face-touch while the maximum number of frames labeled as face-touch per video is 8.875.

We made publicly available[2] all the following data:

- Face-touching behavior annotations corresponding to the original video frames;
- Pose estimations, including face and hand key-points detection, obtained by applying OpenPose [17] to all of the original video frames;

[2] github.com/IIT-PAVIS/Face-Touching-Behavior

- Region of interests containing the face (the x and y coordinates of left-top corner of a bounding box, the width and height of it) calculated for all the original video frames, as described in Section 4.4;
- Cross-validation splits used to test methods described in Section 4.4.

4.4 Experimental analysis

In this study, we compare face-touching behavior detection performance of several methods using the presented annotations. These methods are categorized as:

- *i)* A rule-based approach;
- *ii)* Supervised learning with hand-crafted features;
- *iii)* Feature learning with a deep neural network.

All methods exploit the body poses estimated by applying *OpenPose* [17]. The first approach (Section 4.4.1) is fully unsupervised and applies heuristics based on the estimated locations of the hands and face, as well as the estimated size of these body parts. The second set of approaches (Section 4.4.2) includes hand-crafted features, e.g., the transformed locations of the hand key-points and face parts and applies classifiers: Support Vector Machine (SVM) and Neural Network (NN). The last method (Section 4.4.3) involves fine-tuning a pre-trained ResNet152 when the automatically extracted region-of-interests covering the face of a person are the inputs.

We split our dataset into five non-overlapping folds. This division was made in the way that, each fold has similar amount of images belonging to face-touch class. These splits were used to perform 5-fold cross validation such that four out of five folds were used for training while the remaining fold was used for testing. It is important to highlight that a person whose data was used in the training, was not included for the corresponding testing. In other words, all approaches were applied within a person-independent setting. As the evaluation metrics, we used $F1_{score}$ and Matthews Correlation Coefficient (MCC) as defined in Section 1.2.2 and reported here for convenience:

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{4.1}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}}, \tag{4.2}$$

where Precision=TP/(TP+FP), Recall=TP/(TP+FN) and TP, TN, FP and FN stand for True Positive, True Negative, False Positive and False Negative, respectively.

The positive class represents face-touch behavior while the negative class represents no-face-touch.

4.4.1 Rule-based detection

This method relies on three bounding boxes, which cover the face and the two hands. Once we extracted the full-body pose of a person by applying OpenPose [17], the estimated image positions of nose, eyes and ears were used to define a bounding box covering the face. The image position of the nose was used as the center of the bounding box. In case of a failure in nose detection, the bounding box was not generated and the corresponding frame was classified as no-face-touch. Otherwise, if the locations of both of the ears were estimated, the width of the bounding box was calculated from one ear to another. In case the left ear was not detected, the width of the bounding box was calculated as the difference between the right ear and the left eye. If the right ear was not detected, the width of the bounding box was calculated as the difference between the left ear and the right eye. When both of the ears were not detected, the bounding box width was calculated as the distance between the eyes. The height of the bounding box was set equal to the bounding box width calculated.

The bounding boxes covering the right hand and the left hand individually were detected similarly. Given that OpenPose results in 21 key-points per hand, we used their image position predictions to define a bounding box, which tightly covers the hand. In the optimal case, all key-points are detected and the hand is totally being covered. However, in case some

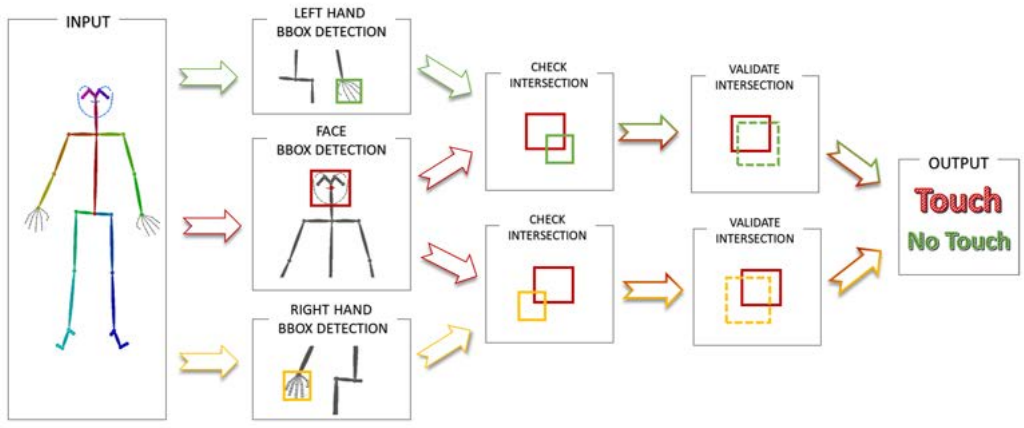


Figure 4.3 Schematic representation of the rule-based pipeline for face-touch detection.

key-points were not detected, we used only the detected ones. In case none of the key-points of both hands were detected, then the corresponding frame was classified as no-face-touch.

During the tests on our dataset [112], the number of frames classified as no-face-touch due to not being able to detect the face or both hands happened only for 8.039 frames, which is very few given that in total we have more than 2M frames having no-face-touch label.

Given a bounding box for face and bounding boxes for hands, firstly we determined whether the bounding box of right or left hand was overlapping with the bounding box of the face. In case both were not overlapping, the corresponding frame was classified as no-face-touch. Otherwise, we calculated the area of the face bounding box and the area of the hand bounding box that was overlapping with the face bounding box. If the area of the face was bigger than the area of the hand, then the corresponding frame was classified as face-touch, otherwise, it was classified as no-face-touch. Herein, we assumed that the face is bigger than the hand when the hand touches the face. With that assumption, we target to eliminate possible false positives, which arise when the hand is in front of the face with respect to the camera position (e.g., when the hand is approaching the face) but not touching the face. Schematic representation of the code is given in Fig. 4.3.

Some example qualitative results of this rule-based approach are given in Fig. 4.4.

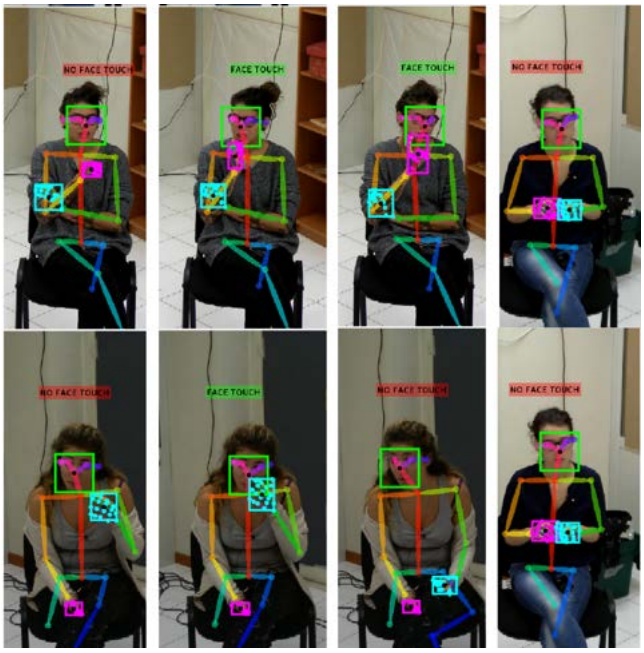


Figure 4.4 Green, magenta and cyan bounding boxes are estimated, corresponding to face, right hand and left hand, respectively. Black dots represent the center of the bounding boxes. The magenta and cyan dots are the key-points belonging to right and left hand, detected by OpenPose [17].

Algorithm 1 Rule-based face touch detection pseudocode

```
def detect_face_bbox:
    if not nose_pos:    return False

    if left_ear_pos:    left_pos = left_ear_pos
    elif left_eye_pos:  left_pos = left_eye_pos
    else:   return False

    if right_ear_pos:    right_pos = right_ear_pos
    elif right_eye_pos:  right_pos = right_eye_pos
    else:   return False

    face_width = | right_pos — left_pos |
    face_height = face_width

    return create_bbox(width=face_width, height=face_height, center=nose_pos)

def detect_hand_bbox:
    if len(detected_hand_keypoints) == 0: return False

    bbox_max_x = find_coordinate_max_x(detected_hand_keypoints)
    bbox_min_x = find_coordinate_min_x(detected_hand_keypoints)
    bbox_max_y = find_coordinate_max_y(detected_hand_keypoints)
    bbox_min_y = find_coordinate_min_y(detected_hand_keypoints)

    return create_bbox( max_x=bbox_max_x, min_x=bbox_min_x,
                        max_y=bbox_max_y, min_y=bbox_min_y)

def face_touch_found:
    if  not detect_face_bbox():     return False

    if  intersection_exist(detect_face_bbox(), detect_hand_bbox(left_hand_keypoints)):
        if area(detect_face_bbox()) > area(detect_hand_bbox(left_hand_keypoints)):
            return True

    if  intersection_exist(detect_face_bbox(), detect_hand_bbox(right_hand_keypoints)):
        if area(detect_face_bbox()) > area(detect_hand_bbox(right_hand_keypoints)):
            return True

    return False
```

4.4.2 Hand-crafted features-based detection

This Section describes the hand-crafted features, the Linear Support Vector Machine and the shallow Neural Network used for training and inference.

Face parts and hand key-points. The image location of nose, eyes and ears (in total 5 features), as well as 21 key-points for each hand, were estimated using OpenPose [17]. These locations (in total 94 features) were then transformed with respect to the location of the hips center and the absolute values of the transformed locations were used for training and inference of the classifiers.

Face and hands bounding boxes. The face and hands bounding boxes were obtained as defined in Section 4.4.1. Their center locations are transformed as applied for face parts and hand key-points. In that way, 3 features from face bounding box (x and y position of transformed center and the width) and 4 features from each hand (x and y position of transformed center, the width and the height of the bounding box), for a total number of 11 features, were extracted.

Linear Support Vector Machine (SVM). We applied balanced training (i.e., under-sampling [122]) such that for each video in the training set, we used all the frames labeled as face-touch and randomly selected the same amount of frames labeled as no-face-touch. The parameters of SVM were chosen based on the best performance obtained in the validation set, which was 20% of the total training set in a cross-validation fold. The validation set was randomly selected and not overlapping with the training set.

Neural Network (NN). We defined a fully connected neural network (Fig. 4.5) having three dense layers with rectified linear units (ReLU) activation function and a final classification layer with Softmax. For regularization purpose, a drop out layer has been inserted after each dense layer. There is no extra pre-processing step on input data except feature normalization, which was applied by dividing the maximum value within each feature to make the range between 0 and 1. The hyperparameters were found based on the best performance of the validation data. We tested:

- Number of neurons, set to be equal in all the dense layers, as $1\times$, $1.5\times$, $2\times$ or $3\times$ the number of features;
- Learning rate as 0.001 or 0.01;
- Dropout percentage as 30% or 50%;
- Batch size as 2.048.

The final network was trained on complete training data, once the optimum hyper parameters were selected. The cross entropy loss was minimized using Adam optimizer.

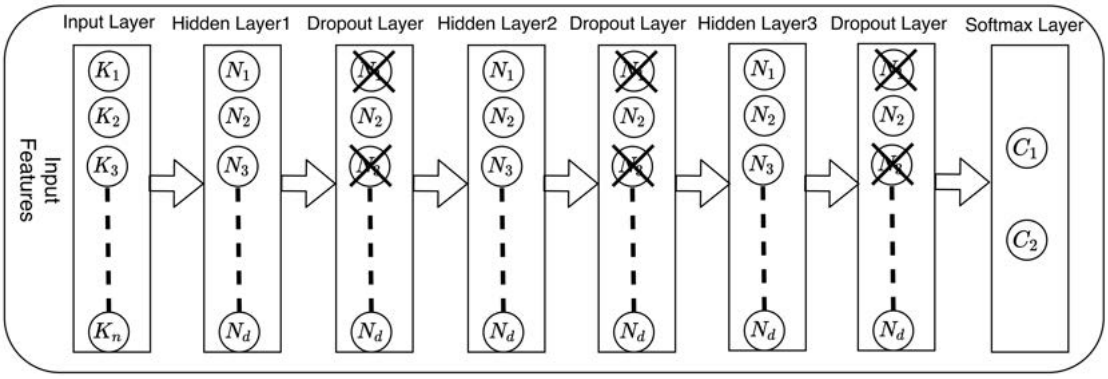


Figure 4.5 The architecture of the applied neural network.

4.4.3 Feature Learning-based detection

The center of the face bounding box (detected as described in Section 4.4.1) was used as the reference central point to crop a 300×300 image from each video frame. The obtained regions of interest were used to fine-tune a ResNet152 Network (pre-trained on ImageNet dataset) for detecting face-touching behavior. We applied under-sampling, such as described in (see the definition in Section 4.4.2, to both training and validation set. Random horizontal axis flipping was applied as data augmentation. The pre-trained weights have been used for the network initialization and we updated all the model parameters during the fine-tuning. We iterated gradient descent for 40 epochs and the validation set was used to select the best performing model.

4.5 Results

The performance of the proposed approaches are given in Table 4.2 in terms of F1-score and MCC, as defined in Section 1.2.2. These results are the average performance of 5-fold cross-validation. Additionally, we report the performance of random guess, which shows the difficulty of the task given a very imbalanced dataset. Random guess was computed by randomly generating a label (face-touch or no-face-touch) for each frame in a test set and then calculating the F1-score and MCC. For each test fold, we calculated 1.000 times the random guess and took the average of the corresponding performances.

Table 4.2 The best performance of each approach in terms of F1-score and MCC. The best result obtained for each metric is emphasized in bold-face. SVM and NN stand for Linear Support Vector Machine and Neural Network, respectively.

	F1-score	**MCC**
Random Guess	6.67	0.0001
Rule-based	75.79	0.76
Hand-crafted features-based		
Face parts and hand keypoints-SVM	55.26	0.56
Face and hands bounding boxes-SVM	46.58	0.50
Concatenation of all-SVM	57.02	0.58
Face parts and hand keypoints-NN	78.75	0.78
Face and hands bounding boxes-NN	69.73	0.69
Concatenation of all-NN	79.92	0.79
Feature learning-based	**83.76**	**0.84**

Fine-tuning ResNet152 performed the best out of all methods. The second best performance was achieved by NN when the concatenation of the two sets of hand-crafted features was used. Out of the two types of hand-crafted features described, face parts and hand key-points showed superior performance as compared to face and hand bounding boxes for both NN and the linear SVM. Although being simple and not requiring training, the rule-based approach still performed better than linear SVM. All methods had higher true negative rates (i.e. correct not-face-touch detection) than the corresponding true positive rates (i.e. correct face-touch detection). The difference between the true negative rate and the true positive rate is bigger than 10% for NN and SVM, while other methods performed more equally for the detection of both classes.

4.5.1 Qualitative results of Feature Learning-based approach

We used the Grad-CAM method [123] to provide visual explanations of the fine-tuned ResNet152 (i.e., the final convolution layer of Resnet152 and the class-specific gradient information) when applied to our dataset [112]. Some results are given in Fig. 4.6. In particular, for the face-touch class, the model focused on the hand(s) that is on the face, no matter what its location was, e.g., on lips, cheek, chin, forehead, etc. For the not-face-touch class, it is harder to say that model focused on a single body part. But, activation maps usually occurred on the face or hand(s). This also shows that the model managed to understand that it is fundamental to focus on hand(s) to differentiate the two classes from each other.

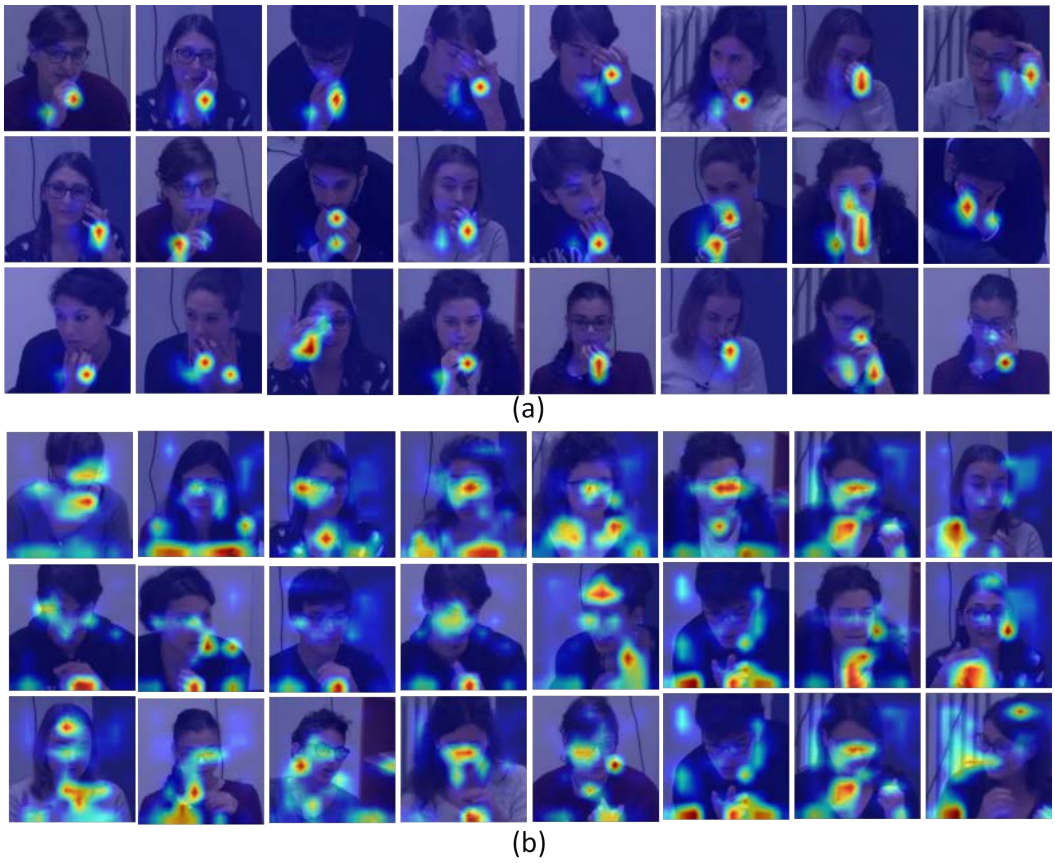


Figure 4.6 Visual explanations of face-touch detection for dataset [112] obtained by using Grad-CAM method [123] for the fine-tuned ResNet152. Activation maps were plotted for (a) face-touch class and (b) not-face-touch class. Note that red regions correspond to a high score for the class.

As an additional analysis, we applied the trained ResNet152 to a different dataset, Cam3D [111], and we obtained the visual explanations of face-touch detection using Grad-CAM [123]. Since the face-touch annotations are not publicly available for Cam3D dataset [111], we did the labeling. It is important to highlight that ResNet152 training was performed on our dataset [112] and the trained model was applied in testing to Cam3D dataset [111]. The images of the two datasets are different in terms of both image resolution and content. In detail, the ResNet152 training was performed with images having lower resolutions (300×300) covering the upper body area, up to the chest, whereas the test images are larger and they include full upper body, background and other objects in the scene, e.g., camera. Therefore, before applying Grad-CAM to Cam3D dataset, the original images (640×480) were resized to 300×300. The corresponding results are given in Fig. 4.7.

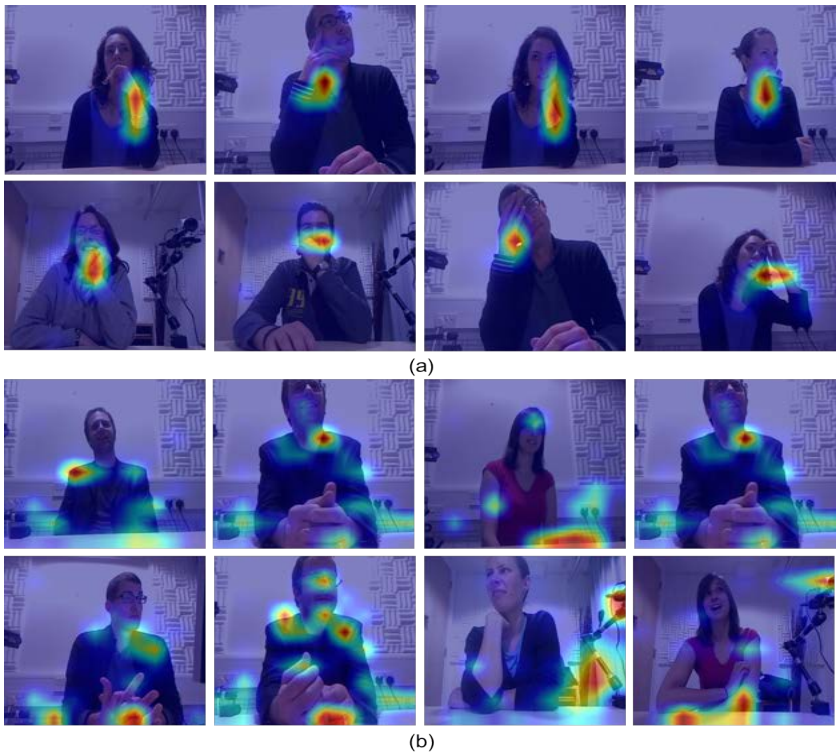


Figure 4.7 Visual explanations of face-touch detection for Cam3D dataset [111] obtained by using Grad-CAM method [123] for the fine-tuned ResNet152. Activation maps were plotted for (a) face-touch class and (b) not-face-touch class. Note that red regions corresponds to high score for class.

For the face-touch class, Fig. 4.7 shows similar results to the ones obtained for our dataset [112], i.e. the model focused mostly on the hands. However, for the not-face-touch class, activation maps could appear in other parts of the image, although the model mostly focused on image locations close to the face and to the hands. These results clearly demonstrate that the fine-tuned ResNet152 trained on our dataset [112] generalize well on different datasets, thanks to the large scale and the face-touch variety of the training set.

4.6 Discussions and future directions

In this Chapter, we have introduced face-touching behavior annotations collected for a dataset composed of audio-visual recordings of small group social interactions [112]. Even though previous manual behavior observation studies show that face-touching behavior is frequent, collecting a significant amount of data having various face-touching gestures, especially from natural videos, is still very challenging. Nevertheless, we present a publicly available dataset

[112], which involves the highest number of participants' behavioral data and contains the largest amount of face-touching behavior annotations: 74K images with face-touch and 2M images for no-face-touch.

Using our dataset [112], we applied several methods for the detection of face-touching behavior. Fine-tuning ResNet152 performed the best among all methods. In addition to performing well, the visual explanations of the fine-tuned model applied to our dataset [112] and on the Cam3D dataset [111] proved that the model is able to focus on hands, especially for the face-touch class, and it is able to differentiate face-touch from not-face-touch. The performances of the applied methods can be used as baselines for future studies. In addition to the images and the face-touching annotations for each video frame, we provide the pose estimations, the image locations of the hands and face, the regions of interest used to fine-tune the CNN architecture and the data splits, together with the code of the baseline methods.

We believe that the supplied modalities can be used in various ways, not only to detect the face-touching behavior but also for forecasting this behavior. As the supplied data is spatio-temporal, a data processing schema involving the temporal dimension of the data can improve the detection performance. Once an accurate detector is trained using this dataset [112], it can be deployed, e.g., for counting or preventing the face-touching behavior in real-time. Such applications can be very useful to restrict this mandatory behavior during pandemic outbreaks. Furthermore, the supplied face-touching annotations can be refined to be used for internal states/emotional expressions classification (e.g., in [124; 125; 126]). Another thing to study can be analyzing the face-touching gestures for the prediction of personality traits [127; 128], dominance [129] and emergent leadership [112; 118; 130].

Chapter 5

AI and Covid-19: The Challenge of Visual Social Distancing

5.1 Introduction

The recent worldwide pandemic emergency raised attention on daily-life circumstances which previously were not a cause of concern. Among them, there are constraints on the physical distance between people as an effective measure to reduce the virus spread. However, beyond the enforcement of regulatory rules, a critical issue for safety is to verify and quantify the actual compliance of people with these restrictions which indeed have a substantial impact on our social life [131]. To this end, a lot of solutions have been proposed [132]. However, cameras in video surveillance settings offer arguably a more viable infrastructure to control the so-called *Social Distancing* (SD).

Visual Social Distancing (VSD) [133] is a particular case of the SD estimation problem, in which the inter-personal distance is estimated from a single uncalibrated image or video. VSD solutions use camera networks that are often deployed in pre-existing video surveillance settings. This allows fast integration of VSD to increase the safety of the population, by detecting recurrent SD violations or by generating statistic analysis. This information can be used to identify risky areas that are subject to crowding and to redefine the architectural design to improve safety in public and private places.

However, the estimation of interpersonal distances from images has at its core a severely ill-posed geometrical problem, as in practical and unconstrained scenarios it is hard to extract robust metric references to measure such distance[134]. This problem is further exacerbated by the fact that most camera networks installed worldwide are not calibrated or, in the worst cases, even intrinsic parameters might be difficult to access. These aspects restrict the use of

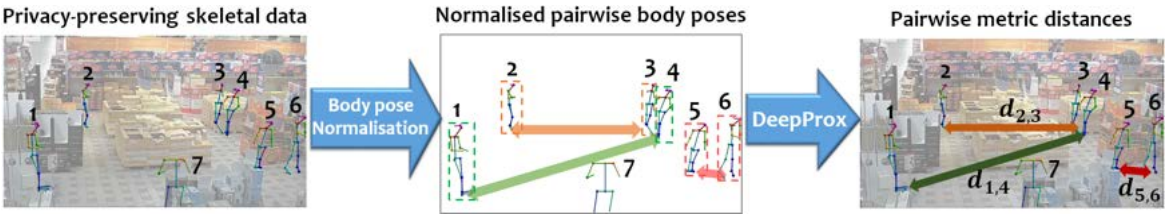


Figure 5.1 Our proposed method *DeepProx* estimates the pairwise metric distances using only 2D skeletal inputs of any pair of detected persons (each pair is indicated by a different color) without the need of pre-calibrating cameras.

proxemics in general and widespread scenarios unless a manual intervention and tuning of parameters is done by an external user.

In this Chapter I present*DeepProx*[1], an approach to estimate interpersonal distances from a single uncalibrated image (see Figure 5.1 for a graphical description). The method exploits the rich structural information encoded in the body joints, whose length and relative position are used to train an end-to-end network to infer the metric distance among any arbitrary pair of people captured by any uncalibrated cameras. Remarkably, we show that the 2D joints do contain enough information for a network to learn and solve such task, despite its intrinsic ambiguity. A key property of *DeepProx* is its ability to generalise to novel camera viewpoints that are not seen at training time. We propose here for the first time a self-calibration loss that coupled with a gradient reversal layer can achieve a favourable invariance to changes in viewpoints. Finally, the design of our pipeline encourages privacy-safe implementations by removing the image content immediately after the pose detection step, practically removing any visual information about the people in the scene. Thus, our network can be trained on anonymous 2D joints positions and similarly at inference time.

DeepProx is trained and evaluated against baselines on public dataset that are upgraded to provide metric distances among people in the scenes. The evaluation setup is explicitly designed to validate the design choices of the approach and to benchmark the the cross-view generalisation capabilities of the proposed method. Additionally, we applied our method to evaluate its capability for monitoring *Social Distancing (SD)* violations in real-world environments in comparison with recent solutions [135; 136]. To summarise, the main contributions of this work are:

[1]This Chapter has been submitted as: "*End-to-end pairwise human proxemics from uncalibrated single images*", Conference on Computer Vision and Pattern Recognition (CVPR 2021) by P. Morerio, M. Bustreo, Y. Wang, A. Del Bue and part of the results illustrated have been included in the paper "*Single Image Human Proxemics Estimation for Visual Social Distancing*", Winter Conference on Applications of Computer Vision (WACV 2020) by M. Aghaei, M. Bustreo, Y. Wang, G. L. Bailo, P. Morerio, A. Del Bue.

- We propose the first uncalibrated end-to-end approach to estimate pairwise interpersonal distances on a single image;
- We introduce a self-calibration loss which regularises training by promoting a branch of network to explicitly learn the key parameters of camera extrinsics;
- We improve *DeepProx* invariance to novel viewpoints using a domain adaption technique adapted to the geometrical problem of interpersonal distance estimation;
- We provide a novel privacy-preserving dataset recorded with multiple cameras in an office environment during 3 months of 24/7 continuous operation.

5.2 Related work

The estimation of interpersonal distances from an image requires the solution of an ill-posed geometrical problem, i.e. the inference from a single view of the 3D spatial arrangement of people in a metric reference space. This Chapter focuses on the geometrical aspects of the problem, as the human body pose is an input of our approach. For this reason, we review pose detection methods closely related to our work, but for an extensive survey please check the work of Chen et al. [137]. Next sections provide an overview on how interpersonal distances are estimated in visual proxemics and how these approaches have been recently used to detect social distances violations using cameras [133].

5.2.1 Human proxemics estimation from images

The inference of human spatial arrangements from images is mostly based on the post-processing of the output given by person or body pose detectors. Although there are works inferring the human interactions directly through the 2D detections [138; 139], most studies attempt at finding a geometrical reference such that metric measurements can be computed [140; 141]. When the camera is calibrated, the metric interpersonal distances on the ground plane can be retrieved using an homography [140]. However, when the camera is not calibrated, some assumptions regarding camera parameters and dimensions of body parts, e.g. the size of face, are often exploited to achieve an approximate estimation of the camera viewpoint, thus enabling the inference of people positions in 3D [141]. Differently, *DeepProx* allows to compute distances in a single pass without pre-calibration or additional step for the scene geometry estimation in terms of camera view estimation and common plane identification.

The estimation of multi-person pose in 3D can also contribute to a more discriminative spatial description for the study of human interaction, whereas most recent works exploit deep learning to directly localise multi-person human pose in 3D from a monocular setting [142; 143; 144; 145]. LCR-Net++ [142] first locates human bounding boxes and classify those boxes into a set of K anchor-poses. A regression module is then used to refine the anchor-pose to the final prediction. Instead of explicitly predicting the human bounding box, Mehta et al. [143] propose occlusion-robust pose-maps (ORPM) which outputs a fixed number of maps encoding the 3D joint locations of all people in the scene. Xnext [144] is a three-staged approach to address multi-person 3D pose estimation from monocular video stream in real-time. Moreover, ordinal relations among joint parts can also benefit the accuracy of joint localisation by serving as a weak supervision [145]. Li et al. propose a hierarchical multi-person ordinal relations (HMOR) [145] that further leverages the multi-person interaction relations on the depth level to improve 3D pose estimation. However, in all these works, the estimated 3D poses are often relative to the camera reference (e.g. up to a scale), which is not enough for computing a metric pairwise proxemics distance without reverting to a further camera calibration. On the contrary, *DeepProx* provides a metric pairwise distance directly from 2D data and without requiring a further calibration step as this stage is learned in the proposed network model.

5.2.2 Human proxemics for Social Distancing

The estimation of interpersonal distance is a key aspect for monitoring Social Distancing (SD), potentially helping to safeguard the health of people during the pandemic. Existing solutions attempt to monitor the interpersonal distance between people and such aspect is strongly related to our work. Most of the approaches follow a similar paradigm described in the previous section: First localizing persons on the 2D image [17; 53; 146] and then relating ground floor position distances assuming a planar surface with a known [147] or estimated homography [135; 148].

The system Inter-hommines [148] exploits CenterNet [146] as person detector with a refined localisation of human's feet, while the homography is estimated with a manual procedure based on a fixed pattern of nine markers on the ground. Another system[2] follows a similar approach, while people are detected using YOLOv3 [53] and the homography is estimated by the manual selection of key points on the image plane. Differently, the system proposed in [135] detects the violations of SD using OpenPose [17] for the localisation of

[2]https://landing.ai/landing-ai-creates-an-ai-tool-to-help-customers-monitor-social-distancing-in-the-workplace/

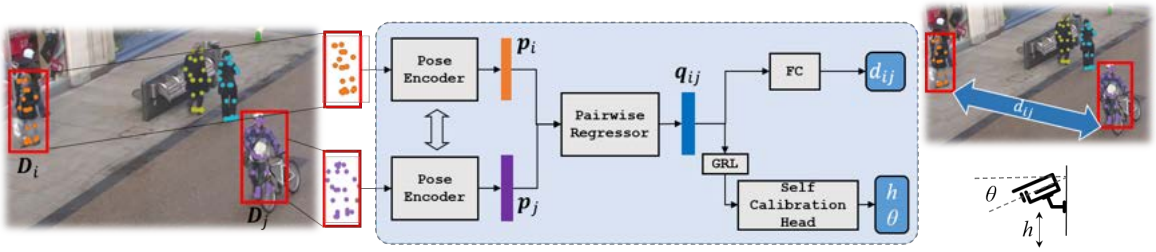


Figure 5.2 Our proposed method *DeepProx* takes as input a pair of 2D human body joints from any image and estimate the pairwise distance. The `Pose Encoder` block first encodes the two sets of 2D joints $\mathbf{D}_i$ and $\mathbf{D}_j$ as feature vector $\mathbf{p}_i$ and $\mathbf{p}_j$, respectively in a higher dimensional space. The two feature vectors are then concatenated and input to the `Pairwise Regressor` block together with a fully connected (FC) layer for learning the pairwise distance. A separate branch after the `Pairwise Regressor` is further introduced to regularise the learning for scene geometry encoding with a gradient reversal layer (GRL) to be more robust to viewpoint variations.

the feet of each detected person and the metric inference uses a semi-automatic homography estimation procedure. The experimental evaluation in [135] shows that using human body pose provides better performance than using bounding boxes. In general, the reviewed approaches for human proxemics estimation in this paragraph cannot function reliably in the wild without either the use of known patterns or human intervention for homography estimation. Differently, *DeepProx* proposes a fully uncalibrated approach, under the assumption of a simplified camera model, without the need of pre-calibrating intrinsic and extrinsic camera parameters for the computation of homography.

Directly estimating the 3D positions of people is another alternative for human proxemics estimation. Although predicting multi-person 3D pose [144; 145; 149] or shape [150] can serve as a proxy, it may complicate the problem as only the position of each person on a common plane, such as the ground, is necessary rather than the 3D positions of all human-body joints. To this regard, Monoloco [136] proposes a feed-forward neural network for 3D human localisation on the ground plane from a monocular camera, which can be applied to monitor SD in a more straightforward manner by computing the pairwise distance using the estimated 3D position of the people in the scene [151]. However, we argue that identifying the 3D position of people in a common ground plane is a redundant step for estimating their pairwise distances. *DeepProx* shows that it is possible to directly infer metric distances only from 2D body poses, without having to previously localise humans in 3D. Moreover, our method do not require any camera calibration during the inference, while Monoloco uses the camera intrinsics to preprocess the input 2D poses.

5.3 Proposed method

The goal of *DeepProx* is to estimate the pairwise interpersonal metric distances from any image taken from uncalibrated cameras as shown in Figure 5.2. The input of our approach are 2D skeletal joints extracted from the image using off-the-shelf body pose detectors. This representation has the advantage of lessening any privacy related issues compared to the direct usage of image content. We show that a pair of 2D poses is sufficient to encode essential information for estimating a metric distance between people through an auxilary task which considers camera extrinsics parameters.

Let $\{\mathbf{D}_i\}_{i\in[1,N]}$ be the set of person detections extracted from a single uncalibrated image, where N denote the total number of detected people. Each detection $\mathbf{D}_i$ is composed of a set of body joints, i.e. $\mathbf{D}_i = \{\mathbf{j}_i^k\}_{k\in[1,K]}$ where K is the number of joints that is specific to the chosen pose detector. Each joint corresponds to a position on the image plane $\mathbf{j}_i^k = \left[u_i^k, v_i^k\right]^\top$. We first pre-process the set of 2D joints, $\mathbf{D}_i$ to normalise them based on the size of each input image (Sec. 5.3.1). Every pair of the joints is then fed to the proposed neural network to regress the pairwise distance (Sec. 5.3.2), where we exploit domain adaptation and generalisation strategies (Sec. 5.3.3) and an auxiliary geometrical loss (Sec. 5.3.4) to be more robust against variations in camera viewpoint.

5.3.1 Body joints normalisation

As mentioned above, we use a human pose estimator to detect a set of joints $\mathbf{D}_i$ for every detection in the images. To prevent the method from overfitting to a specific camera model, it would be beneficial to leverage the 3×3 camera intrinsics matrix $\mathbf{K}$ to normalise the image coordinates as $[x_i^k, y_i^k, 1]^\top = \mathbf{K}^{-1}[u_i^k, v_i^k, 1]^\top$[136]. Yet, this information might be not accessible for most of the cameras, so *DeepProx* considers an uncalibrated scenario. We adopt a pinhole camera model with reasonable assumptions over the intrinsics $\mathbf{K}$ such as: *i)* the skew factor is 0, *ii)* the focal length along X and Y axes is the same, and *iii)* the principal point is at the center of the image such that:

$$\mathbf{K}^{-1} = \frac{1}{f}\begin{bmatrix} 1 & 0 & -\frac{w}{2} \\ 0 & 1 & -\frac{h}{2} \\ 0 & 0 & f \end{bmatrix},$$

where f is the focal length, and w and h are the width and height of the image respectively. This means that, up to a scale factor related to f, we can first normalise the image coordinates

to make the metric inference irrelevant to the image dimension. The normalised homogeneous coordinates $\tilde{\mathbf{j}}_i^k$ are obtained as:

$$\tilde{\mathbf{j}}_i^k = \begin{bmatrix} \frac{1}{w} & 0 & -\frac{1}{2} \\ 0 & \frac{1}{w} & -\frac{h}{2w} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i^k \\ v_i^k \\ 1 \end{bmatrix}. \tag{5.1}$$

We then input a pair of the normalised 2D joint positions $[\widetilde{\mathbf{D}}_i, \widetilde{\mathbf{D}}_j] = \left[\left\{\tilde{\mathbf{j}}_i^k\right\}, \left\{\tilde{\mathbf{j}}_j^k\right\}\right]$ to the network. For joints that are mis-detected by the pose detector, we set their normalised position to zero. As described in the next section and later verified in the experiments, the network is able to implicitly absorb the unknown focal length f in Eq. (5.1) during metric inference of the distance.

5.3.2 DeepProx network architecture

Our model consists of three main blocks, as schematised in Figure 5.2. The `Pose Encoder` block E first projects the normalised 2D joint coordinates onto a higher dimensional space so that, ideally, the encoder should learn how to map into a manifold isomorphic to the 3D world space. Each set of normalised joints $\widetilde{\mathbf{D}}_i$ is then mapped to $\mathbf{p}_i \in \mathbb{R}^m$ where m is the dimension of the projected vector. In such space, the `Pairwise Regressor` block together with a fully connected (FC) layer learns a distance function $F : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ between pairs $[\mathbf{p}_i, \mathbf{p}_j] = [E(\widetilde{\mathbf{D}}_i), E(\widetilde{\mathbf{D}}_i)]$ such that:

$$\hat{d}_{ij} = F(\mathbf{p}_i, \mathbf{p}_j). \tag{5.2}$$

In practice, the pair of projected vectors $[\mathbf{p}_i, \mathbf{p}_j]$ are first concatenated $[\mathbf{p}_i, \mathbf{p}_j]$ and then input to `Pairwise Regressor` block which process them with a stack of fully-connected layers.

We introduce the `Self Calibration` head to provide extra supervision on the network training as detailed in the following subsection. The `Self Calibration` head takes as input $\mathbf{q}_{ij}$, the feature vector output by `Pairwise Regressor` block, process it with fully connected layers and finally regresses two key camera parameters: The height h and tilt angle θ which are commonly used as a realistic approximation of camera extrinsic whenever objects of interest are located approximately at ground level [152].

5.3.3 Promoting viewpoint invariance with Gradient Reversal

When training data is not sufficient to cover all possible camera views, *DeepProx* runs the risk of overfitting the model to specific viewpoints, leading to poor performance when testing on novel views. Intuitively, we would like to have the feature $\mathbf{q}_{ij}$ to mainly represent a canonical view point for the estimation of pairwise distances, thus promoting a view-invariant performance. In other words, we argue that the feature representation $\mathbf{q}_{ij}$ which is used to estimate the pairwise distances should possibly be *independent* from the camera view, for the model to be able to generalise well to unseen geometries. Quite unsurprisingly, we have noticed instead that the network is able to learn all of the camera configurations seen during training. In fact, even if training the model for the only task of predicting $\hat{d}$, we found that the features $\mathbf{q}_{ij}$ retain enough geometric information to allow regressing h and θ with high precision, *which is not what we desire*.

To achieve the purpose of promoting a view-invariant feature representation, we introduce a **Gradient Reversal Layer (GRL)** [153; 154] before the `Self Calibration` block. Originally devised for Domain Adaptation purposes, the GRL layer implements a pseudo-function $R_\lambda(\mathbf{x})$ which simply reverses and scales gradients during back-propagation:

$$R_\lambda(\mathbf{x}) = \mathbf{x}, \quad \frac{dR_\lambda}{d\mathbf{x}} = -\lambda\mathbf{I}, \quad \lambda > 0. \tag{5.3}$$

Such layer was introduced to generalise feature representations learned in a *source* domain to some *target* domain, by making them domain-invariant. Here, an auxiliary domain classifier is trained with the task of predicting from which domain the features are coming from. Instead, by reversing gradients, the GRL is working to confound such domain classifier.

Similarly, in our network, the `Self Calibration` block performs an auxiliary regression task, i.e. it tries to learn how to regress h and θ from $\mathbf{q}_{ij}$. The gradients propagated downwards through the `Pairwise Regressor` and the `Pose Encoder` are instead working against the auxiliary task, being reversed with a minus sign by the GRL layer. As a result, the process will produce a feature representation $\mathbf{q}_{ij}$ which is highly uninformative for the auxiliary task itself (i.e. the inference of camera position and orientation). Being optimised *against* the task instead of *for* the regression task, $\mathbf{q}_{ij}$ is thus deprived of all of the information about h and θ.

Note that setting $\lambda = -1$ in Eq. 5.3 makes our model a standard multitask learning architecture, which learns jointly the two regression tasks. However, let us stress that this is *not* what we desire. The feature representation $\mathbf{q}_{ij}$ would in this case be highly entangled with the camera view, making generalisation to new views more difficult.

5.3.4 Distance and Auxiliary Losses

The full network is trained with the main objective of minimizing the $\ell 1$ norm between the groundtruth pairwise distance d and the predicted $\hat{d}$:

$$\mathscr{L}_{dist} = |d - \hat{d}|_1. \tag{5.4}$$

Additionally, we introduce the auxiliary geometry loss $\mathscr{L}_{aux}$ with the aim of regularizing the distance prediction by explicitly regress the camera parameters h and θ as mentioned above:

$$\mathscr{L}_{aux} = |\theta - \hat{\theta}|_1 + |h - \hat{h}|_1. \tag{5.5}$$

The total loss is given as a weighted sum of the two losses:

$$\mathscr{L} = \mathscr{L}_{dist} + \alpha \mathscr{L}_{aux}, \tag{5.6}$$

where α is used to weight the auxiliary loss, whereas the impact of different α on training the network will be shown in the Sec. 5.4.

5.4 Experiments

We train and evaluate our method *DeepProx* against state-of-the-art methods with a set of publicly available dataset covering both indoor and outdoor scenes, as well as a new privacy-preserving dataset featuring a corporate office area in a 24/7 functioning regime (*Office24/7*). We measure the performance of the method by evaluating the *Mean Absolute Error (MAE)* between the estimated pairwise distances and the ground-truth ones:

$$\Delta d = \frac{\sum_{i=1}^{N} |d_i - \hat{d}_i|}{N}, \tag{5.7}$$

where N is number of samples of pairwise proxemics distances in the test set. We report MAE with the metric unit in meters. Moreover, we also evaluate our method for the problem of detecting the violations of SD from an image, i.e. solving a Visual Social Distancing problem (VSD) [133]. This problem can be regarded as a binary classification task with a SD threshold set to be 2m (e.g. 1m radius from the person centroid on the ground). For the classification, we report the results of all experiments using the *F1* score, which is a standard

in binary classification evaluation:

$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall} = \frac{TP}{TP + 0.5(FP + FN)}.$$

We compare *DeepProx* against three state-of-the-art methods:

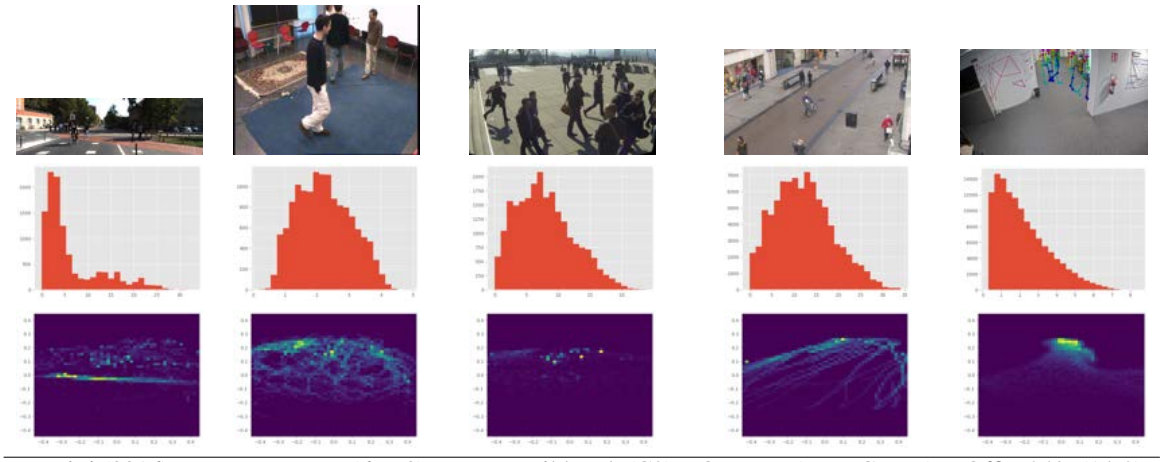
- **Monoloco** [136] is a recent method for monocular 3D humans localisation from a single image which requires camera intrinsics for image normalisation. Interpersonal distances are then computed directly from the 3D localisation results.
- **VSD-HIL** [135] relies on a Human-In-the-Loop strategy to estimate two perspective ratios of the scene for approximation of homography. The method projects a circle of a radius $\sim 1m$ on the ground plane to the image as an ellipse per detected person, and an SD violation is detected if any two ellipses intersect.
- **AutoRect** [155] provides a model for single image automatic rectification by estimating the homography **H** between two image planes. The circular safe space around each detected person on the scene is then projected in the shape of an ellipse, and SD violation is detected if any two ellipses intersect as in [135].

Note that both **VSD-HIL** and **AutoRect** judge the SD violations purely on the image plane without explicit estimation of metric distances. Therefore, we are only able to compare against **Monoloco** in terms of the MAE of the pairwise proxemics distances, while we can compare against all above-mentioned methods in terms of the VSD estimation.

Implementation details. All modules are stacks of fully connected layers followed by batch normalisation and ReLU activations. We found dropout to be detrimental. The `Pose Encoder` is a stack of 3 residual blocks, each composed by 2 FC layers of size 512. The `Pairwise Regressor` and the `Self Calibration Head` have the very same structure, but are double in layer size (1024) and have no skip-connections. Two ad-hoc FC layers with no activation nor batch normalisation allow to regress d and (h, θ).

5.4.1 Dataset

Publicly available dataset: We use real recordings of people in both indoor and outdoor scenes from publicly available datasets, including **EPFL-Mpv** [156], **EPFL-Wildtrack** [157], **Oxtown** [158] which originally used for the evaluation of multi-person tracking (see Figure 5.3 for an overview). These dataset have already been evaluated by [135] for detecting SD violations, in addition here we upgrade the dataset to provide metric interpersonal



Kitti (0016) EPFL-Mpv (6p-c0) EPFL-Wildtrack (C3) OxTown (TownCentre) Office24/7 (14-2)

Figure 5.3 First row shows image samples from some sequences in the dataset. Second row presents dataset statistics as an histogram with the *y*-axis for the number of samples and the *x*-axis for their interpersonal distance in meters. Third row shows a heatmap of the spatial arrangements of people in each sequence (normalised by the image dimension). This dataset covers both indoor and outdoor scenes and contain sequences captured by different camera settings. Interpersonal distances and camera viewpoints encompass a rich variety of scenarios.

distances. The ground-truth pairwise proxemics distance between any two detected people is computed based on their 3D position on the ground plane using the given an homography whenever available. We also increase the range of sequences tested in [135] with **Kitti** dataset [159] with four sequences[3] that are selected because they feature the most number of people for training and evaluating our method. As Kitti provides the 3D annotations of people in the scene, we therefore compute the ground-truth pairwise proxemics distance between any two detected people using their corresponding 3D annotation positions. All distances measured are given in meters.

In addition to Kitti, we also introduce a new dataset **Office24/7** with recordings collected in a time span of 3 months. This dataset has been recorded from three CCTV cameras from different view points to capture the natural behavior of occupants in an office environment. To protect the privacy of people being monitored, the camera system runs the pose detector on the fly to detect people and localise their body joints at 1 fps and then deletes the images. Thus the dataset contains only the 2D joints of detected people in the open area, e.g aisles and coffee area. We provide a background image per camera view, for privacy-

[3]Sequences of id 13, 16, 17 and 19 are selected from the object tracking benchmark.

preserving visualisation (see Figure 5.3) and the calibration matrices of each camera needed for computing the ground-truth.

2D pose detection. We applied OpenPose [17] to all the images of the dataset, and each detected person is represented in the format of 25 skeletal joints. We consider a detection to be valid only if at least one joint from the feet is detected and at least half of the body joints of a person are detected. The detection of feet joints is necessary so that the bottom centers of the detected people lie on the ground plane for the computation of ground-truth pairwise proxemics using the given homography.

Dataset statistics. In total, we have images captured from 16 different camera setups from the publicly available dataset, and additional 3 camera setups from our new Office24/7 privacy-preserving dataset. The dataset presents diverse camera viewpoints (Figure 5.3, first row from left to right): Kitti is a standard street-level viewpoint for an automotive scenario where the camera is installed on the top of the car looking ahead horizontally, with $h = 1.65m$ and $\theta = 0$, EPFL-Wildtrack has seven Go-pro cameras monitoring an outdoor square with a large field of view and a height at shoulder and top-head level ($h \in [1.7m, 3.4m]$ and $\theta \in [-20^\circ, -9^\circ]$), Oxtown provides a standard CCTV scenario with a high camera height ($h = 7.8m, \theta = -20^\circ$) and, EPFL-Mpv features an indoor scenario with four cameras placed slightly above people head, however we are not able to retrieve the exact value of camera height and tilt angle as the camera extrinsics are not available, finally, Office24/7 has three roof-level indoor cameras with remarkable perspective distortions ($h \in [2.65m, 2.77m]$ and $\theta \in [-40^\circ, -25^\circ]$).

We present in the second row of Figure 5.3 the distribution of pairwise interpersonal distances over different sets of sequence captured by one examplar camera setup. Depending on the size of field of view and scenario, there is a noticeable diversity over different sequences in terms of the range of pairwise interpersonal distances. Overall, the pairwise distance ranges from 0-35m considering all the images in the dataset. Moreover, we present the spatial arrangement of detected people on the image at the third row of Figure 5.3. The diverse spatial distributions provide additional challenge for estimating the proxemics distances.

Overall we have collected $\sim 270k$ samples of pairwise distances from 16 different sequences. In general, we employ a cross-sequence validation strategy, by iteratively leaving one sequence out for validation, while training on the remaining 15. This strategy is chosen for testing the generalisation capabilities of the proposed model to different camera viewpoints. The number of samples for each camera setup is balanced to $\sim 20k$ during the training and validation. In addition, we use $\sim 400k$ samples from our *Office24/7* dataset to test further

the generalisation of *DeepProx* to new camera viewpoints (i.e. this part of the dataset was never used for training).

5.4.2 Results Discussion

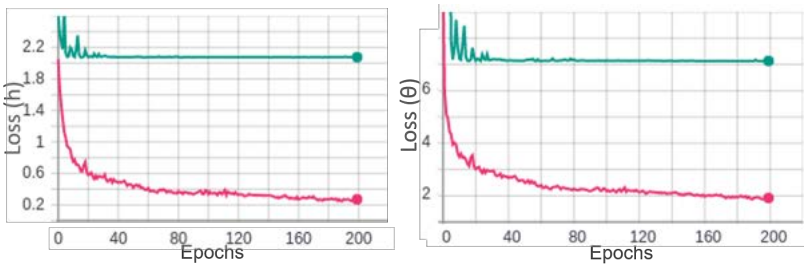


Figure 5.4 Auxiliary losses for h (*left*) and θ (*right*). When $\lambda = 0$ in the GRL (*red*), the `Self-calibration Head` is able to extract view-related information retained by $\mathbf{q}_{ij}$. On the contrary, when $\lambda = 1$ (green), the average errors are higher (and in practice not decreasing) since $\mathbf{q}_{ij}$ is not providing information on h and θ.

Learning camera view-independent features. As mentioned in Sec. 5.3, we posit that features $\mathbf{q}_{ij}$ still retain information on the camera views, even if the `Pairwise Regressor` and `Pose Encoder` blocks are trained with the only task of regressing the pairwise distances. In order to verify such claim, we train the *full* model (i.e. with the *total* loss - Eq. 5.6) by setting $\lambda = 0$ in the Gradient Reversal layer. This modification prevents gradients originated from the auxiliary loss (Eq. 5.5) from flowing through the GRL and consequently through the lower modules, which are thus optimised only using the distance loss (Eq. 5.4). At the same time, the `Self-calibration Head` is trained as always to predict groundtruth h and θ from $\mathbf{q}_{ij}$, but, since gradients are not flowing beyond the GRL layer, *the feature representation is not specifically optimised for the auxiliary task*, but only for the main task of regressing pairwise distance. Nonetheless, $\mathbf{q}_{ij}$ retains enough side information on the camera view to allow the `Self-calibration Head` to minimise its auxiliary loss to a good extent, predicting h and θ with surprising precision as shown in Figure 5.4. Average errors shrink down to $\Delta h \approx 25cm$ and $\Delta\theta \approx 1°$.

On the contrary our aim is to enforce features $\mathbf{q}_{ij}$ to be view-independent for better generalisation. By setting $\lambda = 1$ in Eq. 5.3 we allow gradients to flow through the GRL *with a minus sign*, which prevents the lower modules to encode information on h and θ. Average errors in Figure 5.4 are in fact higher here: $\Delta h \approx 2m$ and $\Delta\theta \approx 6°$.

View-independent features promote generalisation. Table 5.1 reports the MAE of the estimated pairwise proxemics distance over each sequence of the dataset. We report the

Dataset	Seq.	MAE (↓) Monoloco [136]	Monoloco [136] w/ intrinsics	*DeepProx* w/o GRL	*DeepProx* w/ GRL
Kitti	0013	5.05	N.A.	**0.61**	0.62
	0016	4.47	N.A.	1.22	**0.94**
	0017	4.19	N.A.	**0.49**	0.56
	0019	4.44	N.A.	0.89	**0.88**
	Avg.	4.54	N.A.	0.80	**0.75**
EPFL-Mpv	6p-c0	0.88	N.A.	**0.24**	0.31
	6p-c1	1.04	N.A.	0.26	**0.22**
	6p-c2	0.94	N.A.	**0.21**	0.22
	6p-c3	1.02	N.A.	**0.15**	**0.15**
	Avg.	0.97	N.A.	**0.21**	0.23
EPFL-wildtrack	C1	5.53	4.87	1.84	**1.76**
	C2	4.56	3.95	1.75	**1.26**
	C3	4.66	3.73	1.05	**1.01**
	C4	3.49	3.20	0.64	**0.59**
	C5	2.47	2.37	0.48	**0.47**
	C6	5.89	5.06	2.04	**1.74**
	C7	2.35	1.94	1.28	**1.00**
	Avg.	4.58	3.59	1.30	**1.12**
OxTown	Town	7.90	6.41	2.16	**1.87**
Office24/7	14-2	2.03	**1.89**	2.28	**1.89**
	16-1	4.50	2.72	2.22	**1.85**
	19-2	1.40	**1.18**	2.04	1.71
	Avg.	2.64	1.93	2.18	**1.81**

Table 5.1 MAE of the estimated pairwise proxemics distance over each sequence of the dataset following the leave-one-out protocol to test the generalisation on the unseen camera setup. Best results are highlighted in **bold** and numbers are N.A. when the intrinsics are not available.

results of our methods with and without GRL following the leave-one-out evaluation protocol to test the generalisation on the unseen camera setup. The results of Monoloco with and without camera intrinsics are also reported for comparison.[4] On average, our proposed method *DeepProx* which does not require any camera calibration is able to outperform Monoloco even when camera intrinsics are provided. Moreover, in most cases, *DeepProx* can better generalise to new camera viewpoints for regressing the interpersonal distances when trained with GRL, i.e. when features are forced to be view-independent. In the case

[4]Monoloco supplies a default matrix **K** if the correct one is not provided by the user.

of EPFL-Mpv, the average MAE achieved by *DeepProx* without GRL is slightly lower than *DeepProx* without GRL, however the difference is marginal, i.e. 2 cm only.

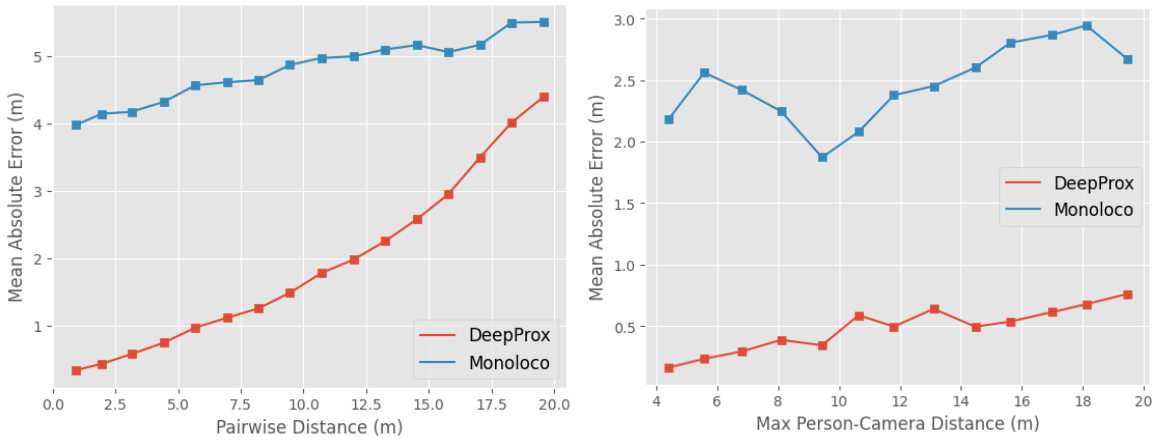


Figure 5.5 The distribution of MAE computed by Monoloco and *DeepProx* on EPFL-wildtrack over the pairwise interpersonal distance (left) and the maximum distance between the pair of people and the camera (right).

Error dependency on pairwise and person-camera distances. As it might be expected, the MAE is higher on the dataset featuring more long-range pairwise distances (i.e. how far people are apart in the scene) such as OxTown, where the average pairwise distance among all people is $\bar{d} \sim 15m$, while it is much smaller on EPFL-Mpv where $\bar{d} \sim 2m$. Moreover, we also expect that pairs of people located far away from the camera will possibly have a larger error in the distance estimation.

Fig. 5.5 verifies these two dependencies by plotting the MAE over the ground-truth pairwise distances (left) and person-camera distances (right) evaluated on the EPFL-Wildtrack dataset. From the left plot, we see that the performance of both Monoloco and *DeepProx* degrade as the pairwise interpersonal distance increases, making evident that estimating proxemic distances when people are far away is a harder task. However, *DeepProx* has a considerably smaller error than Monoloco. The right plot of Fig. 5.5 shows on the x-axis the distance from the camera to all people pairs whose interpersonal distance is $\leq 5m$. We opted for this pre-selection of pairs to reduce the impact of the interpersonal distance variations as seen in the previous plot. The threshold of 5m is set to select pairs close enough to the personal spaces of interest for proxemics studies. From the plot, we can see that the distance between camera and people pairs has a clear impact on the MAE performance, i.e. the MAE increases when people pairs are farther from the camera. *DeepProx* again outperforms Monoloco by a large margin showing its ability to deal with depth variations in the scene by only leveraging pairwise human body poses.

		F1-score (↑)			
Dataset	Seq.	VSD-HIL [135]	AutoRect [155]	Monoloco [136] (+ intrinsics)	*DeepProx*
EPFL-Mpv	6p-c0	77.43 ± 1.12	73.47	73.46 (N.A.)	**89.73**
	6p-c1	71.24 ± 4.92	61.17	70.19 (N.A.)	**90.78**
	6p-c2	75.69 ± 3.87	78.14	77.20 (N.A.)	**90.46**
	6p-c3	73.34 ± 3.51	58.36	72.60 (N.A.)	**93.08**
	Avg.	74.42 ± 3.35	67.75	73.36 (N.A.)	**91.01**
EPFL-wildtrack	C1	**84.78 ± 1.11**	61.80	70.07 (70.50)	79.34
	C2	**83.84 ± 1.19**	57.27	68.31 (68.11)	77.16
	C3	85.52 ± 2.08	45.21	66.73 (67.10)	**86.12**
	C4	81.10 ± 4.03	35.06	64.29 (65.13)	**91.73**
	C5	67.63 ± 3.63	50.99	54.80 (54.69)	**87.80**
	C6	62.87 ± 2.23	39.54	49.64 (49.58)	**80.15**
	C7	**84.46 ± 3.51**	55.63	68.44 (68.74)	80.69
	Avg.	78.60 ± 2.54	49.77	63.18 (63.40)	**83.29**
OxTown	Town	**76.94 ± 4.52**	51.78	54.57 (54.54)	72.86

Table 5.2 Comparison with the state-of-the-art approaches for the evaluation of detection of SD violations in terms of F1 score. Best results are highlighted in **bold** and numbers are N.A. when the intrinsics are not available.

Inference speed at test time. *DeepProx* is very appealing in practical applications since it is extremely lightweight and fast. A single forward of a batch of 190 examples (*i.e.*pairs - corresponding to 20 people in the scene) on an Intel Core i9-9900 CPU @ 3.10GHz takes $\sim 15ms$, which can be reduced to $\sim 3ms$ on a GPU (e.g. NVIDIA GeForce RTX 2080 Ti).

5.4.3 Visual Social Distance (VSD) estimation

In this section we apply *DeepProx* to estimate VSD from single images. We use the estimated pairwise distance to decide if an SD violation occurs. Table 5.2 shows the F1 score of each sequence in the publicly available dataset used by [135] in their evaluation. We observe that *DeepProx* always outperform AutoRect and Monoloco (even with the camera intrinsics). Compared to VSD-HIL, our method *DeepProx* is on average better, apart from on Oxtown for its very different viewpoint, which can instead be easily handled by a human operator in VSD-HIL. .

5.5 Conclusion

This Chapter presented *DeepProx* for the estimation of interpersonal distances from a single uncalibrated image. *DeepProx* promotes a fundamental step for the study of proxemics from

visual inputs and it has a relevant application for assessing VSD violations at distance and in the wild. One key aspect of our method is the introduction of a GRL layer that improves generalisation to variant camera viewpoints and provides a better performance in almost all the tested dataset. This layer can be an useful addition to address any learning-based geometrical problem for which there exists a drop of performance related to novel viewpoints. Future efforts will be put to further improve the robustness of the method for the estimation of proxemics distances with people at far or with severe occlusion by exploiting the pose estimation uncertainties.

Chapter 6

AI for Cultural Heritage: Egyptian Mummy CT Scans Segmentation

6.1 Introduction

Analysis of ancient mummies and their funerary equipment is an important task in archaeological studies, as they represent an extremely well preserved and authentic view of the past. Since half of the 19^{th} century, mummies were scientifically investigated by unwrapping the tape or bandage and removing amulets and jewels from the body [160]. However, this process is devastating and irreversible, therefore it is desirable to avoid it, whenever possible. Hence, it is essential to build a mechanism that can acquire information about what is hidden by the bandages without unwrapping the mummy, and provide this information to the archaeological scholars. 3D volumetric scans obtained from CT devices are the most promising data for these cultural heritage studies [161; 162; 163; 164]. In this work, we propose a novel segmentation algorithm and we apply it to the 3D CT scan of an entire mummy, for digitally unwrapping its bandage and visualize its body and the other present objects (Fig. 6.1).[1]

Even though our task is unique, CT scans are standard data used in medical image processing [166]. Several unsupervised, fully-supervised and semi-supervised methods have been proposed in the literature. However, both unsupervised [167] and fully-supervised [168] methods are not suitable for the problem of 3D mummy CT scans segmentation. In fact, mummy's tissues do not have the same characteristics as typical medical data. Hence, labeled data are not available for training, and it is extremely difficult to get manual annotations since

[1]This Chapter has been published as: "*Weakly Supervised Geodesic Segmentation of Egyptian Mummy CT Scans*", International Conference on Pattern Recognition (ICPR 2020) by A. Hati, M. Bustreo, D. Sona, V. Murino, A. Del Bue.

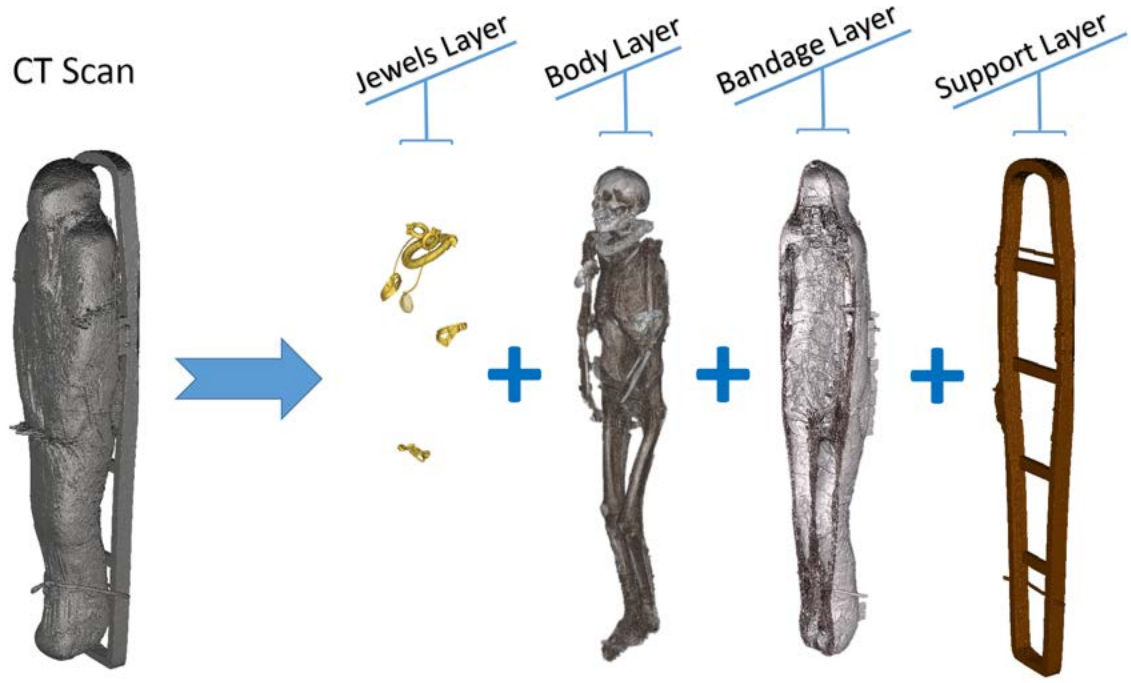


Figure 6.1 Mummy CT scan [165] and its segmentation in *Exterior objects of support*, *Bandage*, *Body* and *Metal* obtained by applying the proposed method.

it is typically a laborious and time-consuming activity, requiring an expert. Furthermore, the mummy's body region is not the most salient [169] part of the CT data, making its segmentation very difficult using unsupervised techniques. Because of all these reasons, to solve the mummy segmentation problem, we propose a weakly supervised method which requires less user interaction than fully supervised methods, while achieving higher accuracy and robustness compared to unsupervised methods [170].

In this Chapter, given a CT scan of a mummy, we segment the data into multiple volumetric semantic parts including bandage, body, metal (*e.g.* jewelry) and other exterior objects, such as the one depicted in Fig 6.1. To accomplish this task, we propose a two-stage weakly supervised algorithm:

1. In the first stage, after detecting the regions exterior to the mummy using prior knowledge about its contents, *e.g.* air and wooden support, we use geodesic distance measure to obtain an approximate segmentation of the mummy into body and bandages, in an unsupervised framework.

2. In the second stage, we apply GrabCut and segment tracking methods to obtain the final segmentation. During this stage, we make use of human interaction in the form of scribbles for refining the result.

In Fig. 6.2, we show a qualitative example of our input data, *i.e.* the scan of an ancient human mummy from coronal, sagittal and axial viewpoints. A detailed description of the input data is reported in Section 6.3.1. It is evident from Fig. 6.2 that at some locations, the bandage tissue is adjacent to the body while both having very similar radiodensity, which makes the problem particularly challenging.

To summarize our contribution, we propose an efficient interactive segmentation method, with limited user interaction needed: user intervention is required only at the final stage of the algorithm, for result refinement. Even though applied to this specific case, our proposed method can be applied to other cases for which annotated data is not available, benefiting from the minimal supervision required. To the best of our knowledge, this is the first work attempting to solve the problem of mummy 3D CT scan segmentation by means of a weakly supervised method.

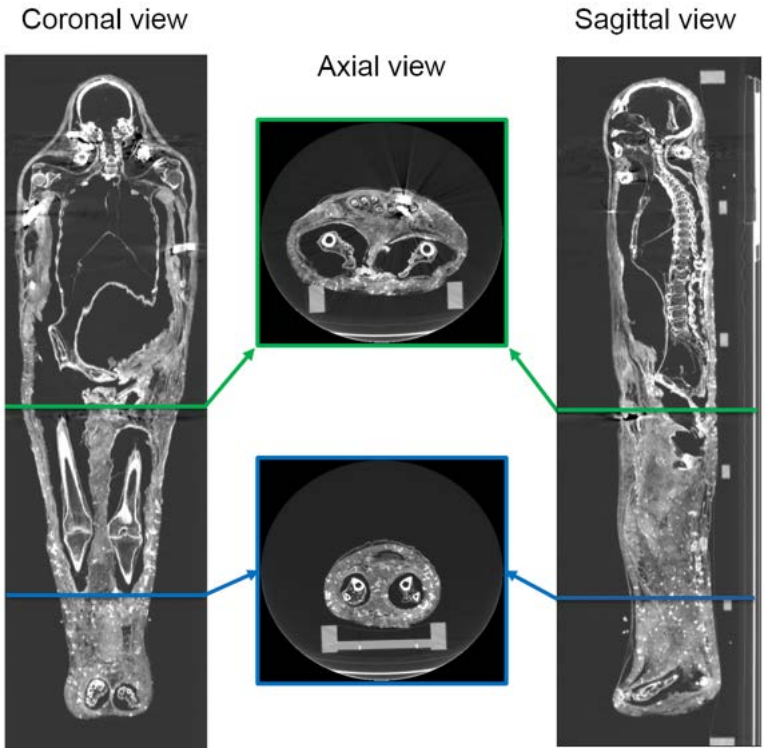


Figure 6.2 Example of coronal, axial and sagittal views of the CT scan of a human mummy [165] (contrast-enhanced for better visualization).

6.2 Related works

In the next Sections, we review relevant works about applications of CT scans to cultural heritage studies and CT scan analysis in medical image segmentation.

6.2.1 CT scan analysis in Cultural Heritage

In the Cultural Heritage field, CT scans are frequently used to analyze archaeological samples. Albertin *et al*. [171] proposed virtual reading and digitization of text in ancient books and handwritten documents from their 3D tomographic images. The presence of metals (*e.g.* iron, calcium) in ancient inks [172] produces sufficient contrast between inked text and the paper in the images. Similar virtual unwrapping of fragile scroll fragments has been explored in [173]. Another collection of works emphasize on obtaining details of archaeological objects' (*e.g.* sculptures) inner structures [174; 175]. This helps in proper planning for their conservation and restoration, as well as understanding historical artifact creation techniques. Typically, a CT volume of the object is reconstructed and visualized for its analysis [176]. Works in [177; 178] used CT scans of mummies to study their general conditions, pathology, embalming and bandaging techniques, as well as to identify the bandaged species in the case of animals. However, the scope of these works is limited to manually studying the 3D visualization. In the given task, we propose a semi-automatic method to segment different regions in the mummy CT data.

6.2.2 CT scan analysis in medical image segmentation

In the medical image segmentation field, one of the most important problems is the segmentation of anatomical structures for medical diagnosis [166]. Noise in the data and variations in data acquisition platforms make it extremely challenging for unsupervised methods to perform well [167]. Recent evolution in supervised deep Convolutional Neural Networks (CNN) [168] overcame these limitations and obtain high-quality segmentation results. CNN-based methods such as U-Net [40], V-Net [179] and DeepMedic [180] have been shown to attain state-of-the-art performance. However, in case of scarcity in training data, training of such networks is hardly feasible. For this reason, semi-supervised methods are preferable. These interactive methods can incorporate task-specific knowledge, thus performing better than automated approaches [181]. Interactions can be provided in the way of clicks [182], contours [183], bounding boxes [184] or scribbles [185]. However, most of these methods require a large amount of user interaction and they frequently focus on segmenting only

specific parts of the body. For example, Rajchl *et al.* [186] used bounding box annotation and solved a conditional random field based energy minimization problem to segment the brain and the lungs. Level set based methods in [187; 188] use manual labeling of regions of interest for segmenting different regions in the brain. These methods apply annotations at the beginning of their respective pipelines and do not have the flexibility to refine the results at later stages. Wang *et al.* [189] proposed a method to segment the placenta and it learns features of the entire organ by propagating foreground-background scribbles drawn on one or a few slices.

6.3 The proposed method

6.3.1 Data representation

The mummy CT scan input data is a volume represented as a sequence of 2D images, where each image is a fixed thickness axial slice (frame) of the mummy. The value of each image voxel represents the measured radiodensity, reported using the Hounsfield Unit (HU) scale. Our goal is to group and segment each voxel into the following regions:

- *Bandage*, which is used to wrap the body;
- *Body* of the mummy, comprised of skin and bone;
- *Metals* (typically jewelry), which lie over the body;
- *Interior hollow space*, which is the gap between the wrapping bandages and the body. This space has been created due to the shrinking of the body over the years;
- *External space*, which is the vacuum around the mummy acquired during scanning;
- *Exterior objects*, such as structures supporting the mummy (*e.g.* wooden support).

In Fig. 6.3a, we show one axial slice of a male mummy with an indication of the regions to be segmented. Fig. 6.3b illustrates the severe artifacts generated by the metals present inside the bandaged mummy, which are not present in standard biomedical data.

6.3.2 Overview of the approach

For segmenting the input CT scan into multiple volumetric semantic parts, we adopt a hierarchical approach and we segment the regions in a sequential manner, as illustrated in

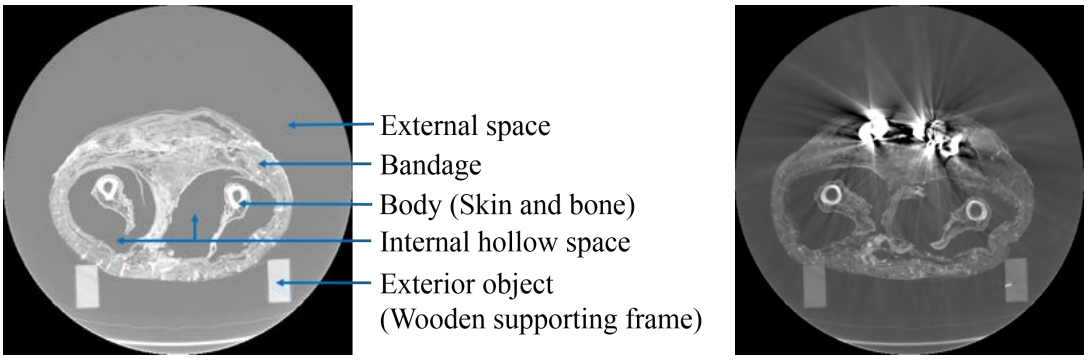


Figure 6.3 Visualization of axial frames of a mummy [165] selected in proximity of the thigh. (a) Some of the regions we are interested to segment are indicated. (b) Artifacts caused by presence of metals.

Fig. 6.4. Given a sequence of 2D images (axial frames), we firstly pre-process every image independently, for identifying all the frame voxels which are not related to the mummy (*i.e. External space*, *Exterior objects* and *Metals*). We achieve this using a combination of histogram equalization, template matching, Hough transform and connected component analysis, as described in detail in Section 6.3.3. The identified set of voxels is then used as a reference in the computation of geodesic distances for obtaining an approximate segmentation of the remaining voxels in *Body* and *Bandage*, as detailed in Section 6.3.4. The final stage of the algorithm consists of a GrabCut based segmentation and in inter-frame segments tracking, for updating and improving the results obtained with the previous stage. Unlike the previous stages where each slice is processed independently, in this final stage, we process the full volume jointly, as described in detail in Section 6.3.5.

In the following Sections, we provide additional details about the implemented method.

6.3.3 Pre-processing

In the pre-processing stage, we aim at segmenting all the elements which are not related to the wrapped mummy's body.

Separating the *External space* (Fig. 6.4, first row) in most of the cases is a straightforward operation, since it is typically composed of air, and therefore it has a known Hounsfield Unit value. In order to ensure a proper segmentation also in the cases in which CT scan values have been scaled or shifted, we use histogram analysis for choosing the threshold better separating voxels with low radiodensity (air) from voxels with higher radiodensity (wrapped mummy or wooden support).

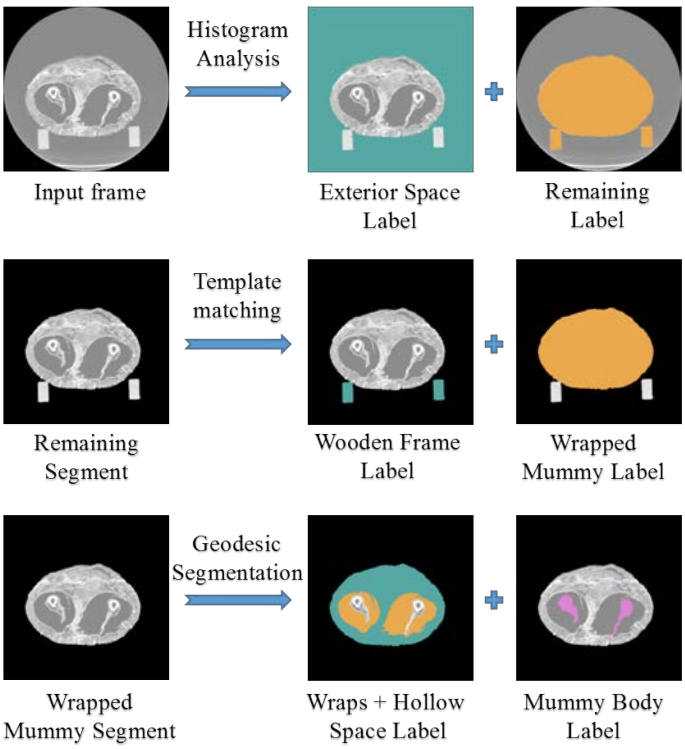


Figure 6.4 Hierarchical segmentation of the mummy's CT scan.

Most of the time, mummies lie on a support (*Exterior object*) which needs to be separated from the body. Mummy supports have regular and known shapes in the CT scans we have been provided. Because of this, we apply template matching for detecting them and optimize the process by locating vertical edges using Hough transform (Fig. 6.4, second row).

Further, it is also known that *Metals* have Hounsfield Unit value larger than any other expected element in the scene. Thus, we find the jewelry present in the mummy by thresholding the voxels and grouping connected components.

The visualizations of the components segmented in this stage are shown in Fig. 6.5 and Fig. 6.12.

6.3.4 Geodesic segmentation

The mummy's *Body* consists of several components (mainly bones and skin, but internal organs can also be observed). Wrapping bandages are non-uniform and their density is not constant, possibly also due to the degradation caused by aging. Hence, the regions we want to segment are non-homogeneous and they are frequently disconnected. Moreover, *Bandage* and *Body* have, overall, a very similar radiodensity and it is not uncommon that two

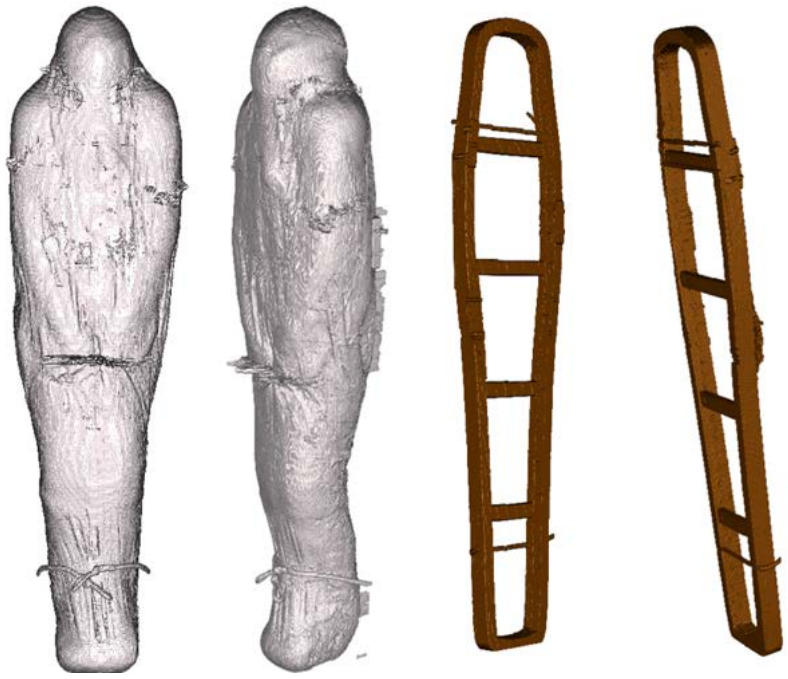


Figure 6.5 Visualization from different viewpoints of the entire wrapped mummy and the support as detected using the proposed method.

adjacent and similar voxels belong to the two different segments. Thus, using the standard Euclidean distance for separating voxels belonging to different regions is not effective. We can, nevertheless, leverage the data structure of the mummy's CT scan: *Body* is wrapped in *Bandage* which is surrounded by the exterior region (*External space* and *Exterior object*), as illustrated in Fig. 6.3a. Hence, we decided to calculate geodesic distance from the exterior region to the unclassified voxels for separating the ones belonging to the *Body* and the ones belonging to the *Bandage*.

Geodesic distance. Consider an undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with set of nodes $\mathscr{V} = \{v_i\}$ and set of edges $\mathscr{E} = \{e_{ij}\}$. Let $\mathrm{P}^{ij} = \{\mathscr{P}_1^{ij}, \mathscr{P}_2^{ij}, \mathscr{P}_3^{ij}, \ldots\}$ denote the set of all paths from node v_i to node v_j that can be traversed in the graph. A path $\mathscr{P}^{ij}$ is a set of edges corresponding to a sequence of node pairs $(v_0, v_1), (v_1, v_2), \ldots, (v_{n-1}, v_n)$ where $v_0 = v_i$, $v_n = v_j$ and n is the length of the path. Given any distance measure $d(\cdot)$ (*e.g.* Euclidean distance), the geodesic distance $d_g(\cdot)$ between a pair of nodes v_i and v_j is defined as the cumulative pairwise distance along the shortest path as follows:

$$d_g(v_i, v_j) = \min_{\mathscr{P}^{ij} \in \mathrm{P}^{ij}} \sum_{(v_k, v_{k+1}) \in \mathscr{P}^{ij}} d(v_k, v_{k+1}), \tag{6.1}$$

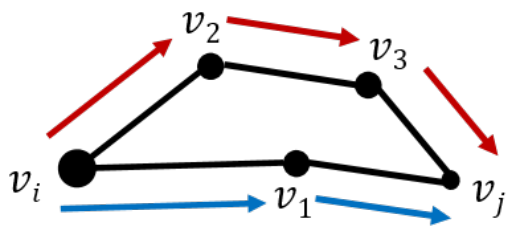


Figure 6.6 Illustration of geodesic distance computation in a graph between nodes v_i and v_j. It is defined as the minimum cumulative distance along two paths indicated in red and blue. $d_g(v_i,v_j) = \min\{d(v_i,v_1)+d(v_1,v_j), d(v_i,v_2)+d(v_2,v_3)+d(v_3,v_j)\}$

and it is illustrated in Fig. 6.6 using an example graph.

In the proposed method, we divided each axial frame into 3×3 patches representing the nodes of the graph and we used the patch average voxel radiodensity as feature for computing the distance $d(v_k, v_{k+1})$ in Eq. (6.1). Note that, instead of patches, it is also possible to use superpixels or regions obtained by any over-segmentation method.

Let us define the set of voxels belonging to the *Body* as R_b, those belonging to the wrapping *Bandages* as R_w and those belonging to either the *External space* or the *Exterior objects*, detected in Section 6.3.3, as R_e. At this stage of the algorithm, we consider the voxels belonging to the *Bandages* and the *Body* together, as part of the same set R_{w+b}. In order to segment R_{w+b}, we first compute average geodesic distance of every node $v_i \in \mathrm{R}_{w+b}$ from the reference region R_e as follows:

$$\bar{d}_g(v_i, \mathrm{R}_e) = \frac{1}{|\mathrm{R}_e|} \sum_{v_j \in \mathrm{R}_e} d_g(v_i, v_j)\,. \tag{6.2}$$

We observe that the reference region contains some noisy voxels with significantly high and low intensity values and they may have a negative influence in the averaging operation in Eq. (6.2). To achieve efficiency and better performance, we modify the formulation in Eq. (6.2) in order to consider only the smallest m geodesic distances $d_g(\cdot)$ between node $v_i \in \mathrm{R}_{w+b}$ and the graph nodes v_j in R_e:

$$\bar{d}_g(v_i, \mathrm{R}_e) = \frac{1}{m} \sum_{v_j \in \mathrm{R}_e} d_g(v_i, v_j) \mathrm{I}(d_g(v_i, v_j) \leq \bar{D}^i), \tag{6.3}$$

where $\mathrm{I}(\cdot)$ is an indicator function and $\bar{D}^i$ is the m^{th} minimum geodesic distance computed for node v_i. For notation simplicity, let us denote $\bar{d}_g(v_i, \mathrm{R}_e)$ as d_i.

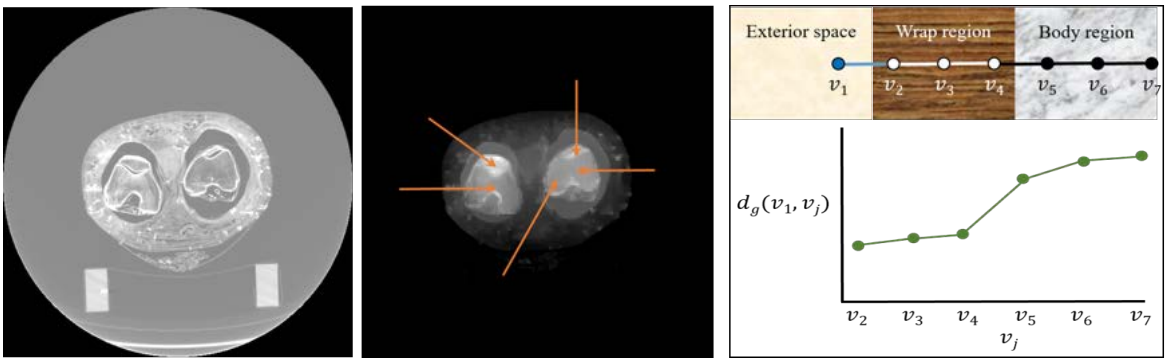


Figure 6.7 Illustration of geodesic distance. (a) Input image. (b) Average geodesic distance $\bar{d}_g$ (Eq. (6.3)) of every voxel from the external space R_e. Arrows indicate that $\bar{d}_g$ increases with traversal from the wrapping bandage region to the body region. (c) Illustration using a synthetic image. $d_g(v_1, v_5)$ is significantly larger than $d_g(v_1, v_4)$ whereas $d_g(v_1, v_4)$ is slightly larger than $d_g(v_1, v_2)$ and $d_g(v_1, v_3)$ because $d(v_4, v_5)$ is larger than remaining pairwise distance $d(\cdot)$.

We argue that patches with smaller geodesic distances constitute the mummy wrap region (R_w) and patches with larger geodesic distances constitute the body region (R_b). This is because computation of geodesic distance between a node pair $(v_i \in \mathrm{R}_b, v_j \in \mathrm{R}_e)$ requires a larger path to be traversed than computing the same for $(v_i \in \mathrm{R}_w, v_j \in \mathrm{R}_e)$. Hence, without loss of generality, we expect a significant change in gradient in the sequence of distances d_i when arranged in ascending order, denoted as $\tilde{d}_i$. It is therefore possible to identify a value m_2 separating the (not necessarily connected) voxels belonging to R_w and R_b:

$$m_2 = \arg\max_i \frac{\partial \tilde{d}_i}{\partial i}, \tag{6.4}$$

$$\mathrm{R}_w = \{v_i \in \mathrm{R}_{w+b} : \tilde{d}_i \leq \tilde{d}_{m_2}\}, \tag{6.5}$$

$$\mathrm{R}_b = \{v_i \in \mathrm{R}_{w+b} : \tilde{d}_i > \tilde{d}_{m_2}\}. \tag{6.6}$$

In Fig. 6.7, we illustrate the described process using a synthetic image and a real axial frame. Note that in most slices, there exists an interior hollow space between the wrapping *Bandage* and *Body*. Detection of this region is trivial since its voxel values are the same as the air voxels in the detected exterior region. In practice, the obtained set R_b may also include some patches from the wrap region in addition to the body patches.

In the next Section, we will describe how to refine these results.

6.3.5 GrabCut segmentation and tracking

In Section 6.3.4, we process every frame in the volume independently obtaining a coarse segmentation of different regions (R_e, R_w, R_b). In this Section, we update this result through volume level processing of data, ensuring information propagation across frames and thus yielding more accurate segmentation results.

GrabCut

Rother *et al*. [184] introduced the GrabCut method for the segmentation of objects of interest in an image. The algorithm separates the pixels into two categories: foreground and background. Typically, a part of the image is hard-labeled by the user as background and the algorithm iteratively assigns labels to the remaining pixels. Here, we apply GrabCut to volumes and we can use the output of the previous processing stage to feed the GrabCut algorithm, removing the need for manual intervention. Specifically, since the goal of this stage is refining the region R_b obtained using Eq. (6.6), we initialize the background region with the set of voxels belonging to $R_w \cup R_e$. The voxels labeled as foreground by GrabCut constitute the updated *Body* region R_b.

However, segmenting the entire CT volume at one go is memory inefficient. Hence, we first split the input scan it into overlapping volumes, each constituted by n_G successive axial slices, and segment these smaller volumes using GrabCut. The obtained results are finally averaged to obtain the final segmentation mask. Using this approach, we ensure that the resulting mask changes smoothly across successive frames.

Tracking

Since the data contains some voxels from different regions with similar appearances, ambiguity in the segmentation is relevant. GrabCut improves the result obtained in Section 6.3.4, but we can further refine it by cross-slice matching of segments belonging to R_b across axial slices. We refer to this process as 'tracking' and the set of matched segments in a certain number of successive slices as a '*track*'. During the process, the segments which cannot be properly tracked are discarded from the final result of R_b and appended to R_w, thus obtaining a more refined output than the GrabCut result.

Let $L^{(k)}$ be the number of tracks in frame-k, $T_i^{(k)}$ denotes the segment corresponding to the i^{th} track in frame-k and $H_i^{(k)}$ denotes an ensemble of segments in this track over past M

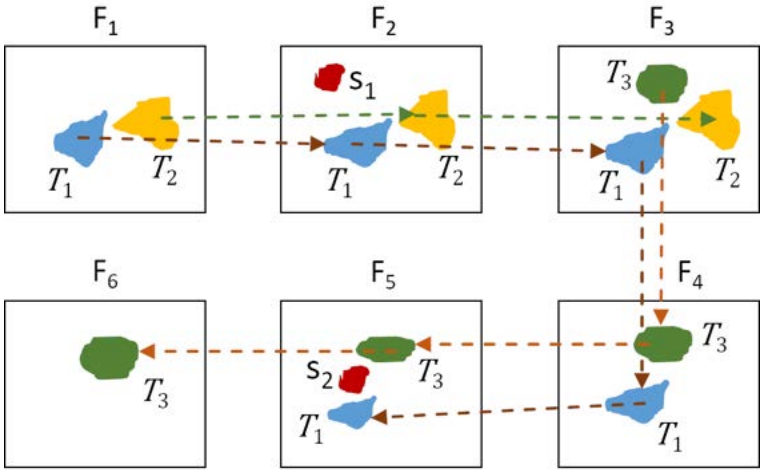


Figure 6.8 Illustration of tracking algorithm. Segments in all the axial slices are detected as the output of the GrabCut stage. User initializes tracks T_1 (*blue* segment) and T_2 (*yellow* segment) in frame-1 and track T_3 in frame-3 (*green* segment). The initialized tracks are expanded across subsequent slices. T_1 and T_2 cease in frame-5 and frame-3, respectively. Tracking algorithm refines the result also removing the spurious (*red*) segments s_1 in frame-2 and s_2 in frame-5.

slices such as:

$$H_i^{(k)} = [T_i^{(k)}, T_i^{(k-1)}, \ldots, T_i^{(k-M+1)}], \forall i = 1,2,\ldots,L^{(k)}. \tag{6.7}$$

In frame-$(k+1)$, a segment s_j is added to the i^{th} track if it has the highest similarity with $H_i^{(k)}$. Thus $T_i^{(k)}$ grows as:

$$T_i^{(k+1)} = \arg \max_{s_j \in R_b^{(k+1)}} \Phi(H_i^{(k)}, s_j), \tag{6.8}$$

where $\Phi(\cdot)$ is the similarity function that computes average similarities between the feature of segment s_j and all segments belonging to a track over past M slices, *i.e.*, $H_i^{(k)}$. Here we consider spatial coordinates of all pixels in each segment as features. Further, the use of $H_i^{(k)}$ instead of $T_i^{(k)}$ while computing $\Phi(\cdot)$ using Eq. (6.8) ensures consistency in matching.

This process begins by a user identifying $L^{(1)}$ valid segments in frame-1 belonging to R_b. These segments initialize tracks $T_1^{(1)}, T_2^{(1)}, \ldots, T_{L^{(1)}}^{(1)}$, and define:

$$H_i^{(1)} = [T_i^{(1)}, T_i^{(1)}, \ldots, T_i^{(1)}], \forall i = 1,2,\ldots,L^{(1)}. \tag{6.9}$$

Subsequently, for the following frames $(2,3,4,\ldots)$, tracks are updated using Eq. (6.7) and Eq. (6.8). Further, a user can, at any intermediate frame, mark a segment to initialize a new track. Any track-i ceases to exist when it does not find any segment with high feature similarity using Eq. (6.8), as:

$$\Phi(H_i^{(k-1)}, \mathrm{s}_j) < \varepsilon, \forall \mathrm{s}_j \in \mathrm{R}_b^{(k)}, \tag{6.10}$$

where ε is a very small threshold. This method is illustrated in Fig. 6.8.

After processing all the frames, segments belonging to the detected tracks jointly constitute the *Body* region of the mummy R_b, which is the most challenging region to separate. The visualization of the segmented body is shown in Fig. 6.9.

Figure 6.9 Visualization from different viewpoints of the body region detected using our method and zoomed visualizations of skull area.

6.4 Results

6.4.1 Parameters

In our experiments, we consider $m = 10$ geodesic distances in Eq. (6.3) for computing average geodesic distance. We verified that the value of m should be chosen in the range $7 \leq m \leq 15$, such that m is big enough to smooth voxels variability but it is not so big to include very large distances in the calculation, which are more likely to be affected by the presence of noisy data.

In the experiments for the GrabCut stage (Section 6.3.4), we choose overlapping volumes of $n_G = 10$ successive frames with stride of 1. The stride of 1 ensures complete overlap between the considered volumes, to avoid any loss of information. The value of n_G should be tuned accordingly to slice resolution: the mummy scan we took into consideration had 3mm axial data resolution. In the case of data with higher resolution, a bigger value of n_G can be chosen, since we can expect that statistics change more slowly across successive frames. Similar reasoning can be applied for the tracking stage parameter choice (Section 6.3.5). We set the parameter for storing track history to $M = 4$. Like n_G, a bigger value can be chosen in the case of higher resolution data.

6.4.2 Dataset

We have tested our approach on the CT scan of a human mummy. In the absence of a large number of subjects for validating the proposed method, we generated additional CT scans by transforming the original scan. For this purpose, we used thin-plate spline mapping [190], where we learn a warping function between two sets of points in the XY-plane and apply it to the coordinates of the input data to transform it.

We first select a set of points $X_1 = (x_1, y_1), X_2 = (x_2, y_2), \ldots, X_n = (x_n, y_n)$ in the XY-plane. Then we slightly perturbed them to obtain another set of points $X'_1 = (x'_1, y'_1), X'_2 = (x'_2, y'_2)$, $\ldots$, $X'_n = (x'_n, y'_n)$ (see Fig. 6.10a). Using thin plate spline, we learn the warping function $f(x,y)$ that maps X_i's to X'_i's with the minimum bending energy [191]. Finally, for each frame in the CT data, $f(x,y)$ is used to map each pixel to the new coordinates and a warped frame is obtained through interpolation of pixels values. This method is particularly flexible since the initial set of points X_i's and their perturbations can be chosen arbitrarily. Hence, in theory, we can obtain an infinite number of warping functions. Fig. 6.10 shows the outputs generated by warping an input frame with two different warping functions.

In the above strategy, every frame in the CT data is transformed using the same warping function. We further investigate the effect of different transformations for every frame. Let N be the number of frames in the CT data and denote X_i's as $X_i^{(0)}$ to X'_i's as $X_i^{(N)}$. For each frame-$k = 1, 2, \ldots, N$, we compute

$$X_i^{(k)} = X_i^{(0)} + k(X_i^{(N)} - X_i^{(0)})/N \tag{6.11}$$

and obtain the corresponding warping function $f_k(x,y)$ using point sets $X_i^{(k-1)}$ and $X_i^{(k)}$. Let us call this set of warping functions as Set 1: $\{f_k(x,y)\}$. To avoid biasing different sections of the mummy with different degrees of transformations, we also obtain three sets by permuting

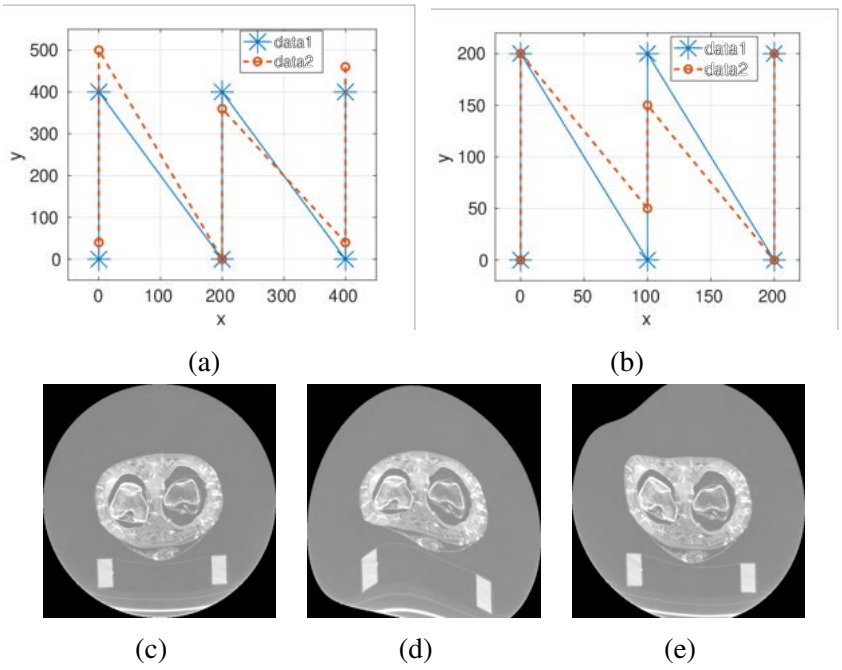


Figure 6.10 Illustration of transformation of a CT data to generate several new data. (a) Two sets of points for which a warping function is computed using thin-plate spline mapping. (b) A different pair of point sets. (c) An axial frame of the CT data. (d,e) The frame in (c) is transformed using the warping functions computed using point sets in (a) and (b), respectively.

$\{f_k\}$ as follows.

$$\text{Set 2: } \{f_N, f_{N-1}, \ldots, f_1\} \tag{6.12}$$

$$\text{Set 3: } \{f_1, f_2, \ldots, f_{N/2}, f_{N/2}, \ldots, f_2, f_1\} \tag{6.13}$$

$$\text{Set 4: } \{f_{N/2}, f_{N/2-1}, \ldots, f_1, f_1, \ldots, f_{N/2-1}, f_{N/2}\} \tag{6.14}$$

6.4.3 Quantitative result

To quantitatively measure the performance of the proposed method, we use intersection over union (IOU) between the predicted result and the manually annotated ground-truth available for the mummy CT scan. IOU has been computed both per frame and on the full mummy volume.

The plot of the IOU score computed for each mummy frame is shown in Fig. 6.11. Poor accuracy for the initial few frames is due to similar visual appearances of the *Bandages* and the *Body* region voxels which cannot be improved using the tracking method. The drop in accuracy near frame-540 is due to the presence of the metallic necklace, which caused

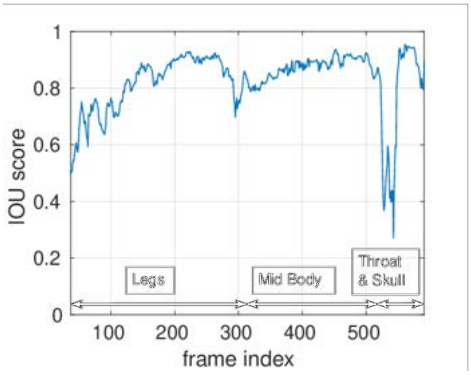


Figure 6.11 Intersection over union (IOU) score computed for every frame. The frames are of feet to the skull of the mummy as we move from lower frame index to higher frame index.

severe artifacts during data acquisition and makes it impossible to recover the correct voxel radiodensity in certain regions.

We analyze the average IOU score on the full mummy volume and on different parts of the body (legs, mid-body and head). Table 6.1 contains the results for the input data and six transformed data obtained using:

1. the warping function of Fig. 6.10a;
2. the warping function of Fig. 6.10b;
3. 4 warping function sets (Eq. (6.11)-(6.14)) obtained from (1).

The ground-truth mask of the input data has also been transformed using the respective warping functions for their IOU score computation. An ablation study of the proposed method is also given in the last row of Table 6.1, which reports the performance of the proposed method without using the tracking algorithm. We observe that the tracking algorithm is indeed beneficial.

Table 6.1 IOU Score on Different Data and Ablation Study.

	Original	Warp 1	Warp 2	Set 1	Set 2	Set 3	Set 4
Legs	0.79	0.77	0.77	0.77	0.77	0.77	0.77
Mid-body	0.87	0.88	0.88	0.88	0.88	0.88	0.89
Head	0.74	0.79	0.81	0.80	0.79	0.79	0.79
Average	0.81	0.81	0.81	0.82	0.81	0.82	0.82
Average-w/o tracking	0.79	0.80	0.81	0.80	0.80	0.80	0.80

Here, we make a comment about the challenges faced. It is a common occurrence that some part of the *Body* and its adjacent wrapping *Bandage* have similar radiodensity. This makes it very difficult to separate them accurately. After transforming the data using warping functions, it is possible that the amount of adjacency between *Bandage* and *Body* with similar radiodensity changes. Accordingly, the performance changes. Further, reflections from metals introduce artifacts in the data by changing (both increasing and decreasing) certain voxel values (Fig. 6.3b), thus providing ambiguous information about the density of different tissues and affecting the performance.

Comparative study: We compare the performance of the proposed method with existing semi-supervised segmentation methods by evaluating them on the mummy scan.

We consider a Graph Cut based clustering method [181] (referred to as *GC*) that had been applied to medical image segmentation. The method learns a Gaussian mixture model for a predefined number of clusters (n_C) and assigns each voxel to one of them. For computing the IOU is chosen the cluster having the highest overlap with the ground-truth mask. Further, to ensure fairness, we provide supervision to the algorithm by additionally marking the exterior region (this supervision is also provided to every method we compare with). In Table 6.2, we report the IOU scores obtained by varying n_C between 3-10 and observe that results do not vary significantly.

Next, we apply the standard GrabCut method [184] (referred to as *GB*) on the input data by drawing bounding boxes around the *Body* region. The achieved performance (Table 6.3) is much inferior to that of our proposed method. This is because our efficient geodesic segmentation output acts as a much better input label for GrabCut. We also investigated how salient the mummy *Body* region is. For this, we have computed saliency map of each frame using the method in [169] (referred to as *SD*). The performance obtained is poor because of the very small contrast between tissues in the *Body* and *Bandages*. Finally, we evaluated a video object segmentation method [192] (referred to as *TIS*) to investigate its applicability to the mummy 2D image sequence. This method uses optical flow and visual saliency of frames, but it is unable to track the mummy's *Body*.

Table 6.2 IOU Score of Graph Cut Based Segmentation Method (GC) [181] Considering n_C Classes.

n_C	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
GC	0.41	0.42	0.41	0.42	0.42	0.41	0.40	0.39

Table 6.3 Comparison of IOU Score using the Proposed Method (*Our*), GrabCut Segmentation (*GB*) [184], Saliency Detection (*SD*) [169], Graph Cut Segmentation (*GC*) [181] and Video Object Segmentation *TIS* [192].

	Our	*GB*	*SD*	*GC*	*TIS*
IOU Score	0.81	0.50	0.46	0.42	0.24

6.4.4 Qualitative result

In this Section, we visualize the produced segments. The pre-processing stage separates the voxels belonging to the wrapped mummy from the external space and the exterior objects (mummy support). Fig. 6.5 shows the result from different viewpoints. Near the fingers, the arms and the neck of the mummy, some scattering can be observed. These are the artifacts caused by the presence of metals in the form of jewelry, as discussed in Section 6.3.1 and highlighted in Fig. 6.3b.

Fig. 6.9 shows the unwrapped mummy's body, including the skull obtained after the final stage of the segmentation pipeline.

In Fig. 6.12, we show the jewelry present in the mummy, segmented in the pre-processing stage.

Though the pipeline is explained using a human mummy, it can also be applied to other kinds of mummies. In Fig. 6.13, we present the results obtained by applying our method on the mummy of a cat.

Figure 6.12 Visualization of metals (jewelry) inside the mummy.

Figure 6.13 Visualization from different viewpoints of the wrapped cat mummy and the corresponding segmented body.

6.5 Conclusions

In this Chapter, we proposed an algorithm for solving the problem of segmenting the CT scan of mummies with minimal supervision. Leveraging the unique structure of the data, we applied geodesic distance measure to obtain an initial segmentation and we further refined it using GrabCut and tracking. In order to validate our method on multiple CT scans, we proposed to use thin-plate spline mapping to transform the original data and artificially generate additional samples. The proposed pipeline yielded very good results using minimal user interactions compared to existing semi-supervised methods. The presence of artifacts in the data affects the segmentation results. Overcoming this difficulty is ongoing research.

Conclusions

Conclusions

Machine Learning, Artificial Intelligence and Artificial Neural Networks have a long story, but, in recent years, they proved to be extremely impactful in solving real-world Computer Vision problems in application to numerous fields, spanning from Medical Diagnosis to Industrial Machine Vision. The recent improvements have been due to theoretical advancements, hardware improvements and available digital data growth. In spite of that, several methodological aspects still need to be clarified and improved to make state-of-the-art algorithms even more powerful and effective.
Given these considerations, I split the thesis into two parts: the first one dedicated to the methodological research I have been involved in during my Ph.D. period, the second one dedicated to the applied research and the related methods I developed.

The demand for a huge amount of annotated data to properly train the Deep Neural Networks causes several issues related to both the difficulties in collecting the data and the costs for expert annotation. In this thesis, I discussed these issues and I proposed some solutions for either collect and label data in a cheaper way or for training models when no training data are present for some of the classes of interest.
I introduced a hardware-based data augmentation approach that can enrich datasets' information not requiring any additional labeling effort. This approach makes use of a custom-designed illumination system that is able to mimic four standard illumination techniques: diffused, dark-field, lateral and frontal illumination. Hardware-based data augmentation can be applied in all the situations in which it is feasible to acquire data in a controlled manner, such as it happens for Machine Vision systems.
For all the scenarios in which this is not possible, I discussed Zero Shot Learning approaches, i.e. the methods tackling the problem of multi-class classification when no training data is available for some of the classes. I proposed to replace the visual embeddings commonly used, which are trained for fully supervised classification, and therefore sub-optimal for the Zero Shot Learning task, with the intermediate representation of an end-to-end captioner that predicts attributes at the instance-level. The proposed enhanced visual embedding, called

VisEn, proved to be able to capture visually-grounded semantic cues, and this was assessed qualitatively and quantitatively. *VisEn* showed to be compatible with any generic Zero Shot Learning method and it can replace default visual embedding, without requiring changes in the pipeline, apart from hyper-parameter tuning. Finally, I proved that *VisEn* systematically improves the recognition performance of three popular approaches.

The huge amount of data required in training is not the only methodological open issue related to Deep Neural Networks. Nevertheless, they can already give a significant contribution in tackling an increasing number of challenges and they can improve state of the art results in many fields. Because of this, I have been involved in several activities focused on addressing some of the current social or cultural-related problems. In this thesis, I chose to illustrate three of these projects, selected to demonstrate the versatility of Computer Vision methods and their applicability to very different contexts. Two of the described projects are related to the 2020 Pandemic Coronavirus outbreak, which suddenly unsettled the global healthcare systems, the world economy and the daily life and work of billions of people. The third project is related to Cultural Heritage preservation.

The recent worldwide pandemic emergency raised attention on daily-life circumstances which previously were not a cause of concern. Among them, reducing risky personal behaviors such as face-touches, and regulation of the physical distance between people, have been indicated as effective measures to reduce the virus spread.

I contributed to preparing and distributing face-touching annotations for a publicly available behavioral dataset. These annotations can be used for training and validating models which are able to detect the face-touching behavior and potentially to forecast and prevent it. In fact, people touch their face frequently, even without realizing it. This aspect is of clear relevance in the current global pandemic emergency as contaminated hands are a potential carrier for the dissemination of respiratory diseases such as influenza and coronavirus. Using the proposed dataset, I applied and benchmarked several methods for the detection of face-touching behavior. Their performances can be used as baselines for future studies.

Related to the 2020 Coronavirus Pandemic, I also presented *DeepProx*, the first approach to estimate interpersonal distances from a single uncalibrated image and I introduced *Office 24/7*, a novel privacy-preserving dataset, recorded with multiple cameras in an office environment during 3 months of 24/7 continuous operation. *DeepProx* uses an end-to-end model that accepts as input two skeletal joints, represented by as a set of 2D image coordinates, and it outputs the metric distance between the corresponding persons. The method has a relevant

application for assessing visual social distancing violations at distance and in the wild. One key aspect of the method is the introduction of a Gradient Reversal Layer that improves generalization to variant camera viewpoints and provides better performance in almost all the tested datasets. The Gradient Reversal Layer can be a useful addition to address any learning-based geometrical problem for which there exists a drop of performance related to novel viewpoints.

Finally, I described an algorithm for solving the problem of segmenting the CT scan of mummies with minimal supervision. Leveraging the unique structure of the data, the algorithm applies geodesic distance measure to obtain an initial segmentation, which is further refined using GrabCut based segmentation and tracking. The proposed pipeline yielded very good results using minimal user interactions compared to existing semi-supervised methods. The obtained segmentations can provide useful information to the archaeological scholars about what is hidden by the bandages, without the devasting process of physical bandaged mummy unwrapping.

Publications

Journal Papers

- Porous pH natural indicators for acidic and basic vapor sensing
  Chemical Engineering Journal, 2020, 126373
  by J. Zia, G. Mancini, M. Bustreo, A. Zych, R. Donno, A. Athanassiou, D. Fragouli

  Abstract: Herein, the development of biobased colorimetric pH porous indicators composed of a polyvinyl alcohol, polyvinyl pyrrolidone and microcrystalline cellulose functionalized with anthocyanins deriving from red cabbage, is presented. As revealed, the presence of 62.5%wt. of microcrystalline cellulose in the composite improves the stability of the porous system in aqueous environments, enhances the porosity, the pores' interconnectivity and reduces the pores' size that in turn significantly improves the responsiveness of the porous indicator to acidic or basic vapors. Specifically, the developed material shows distinct color change when subjected to acidic or basic vapors with a response 6 times faster compared to the indicator without any microcrystalline cellulose content. The color changes are reversible, making possible the use of the same indicator for several times. When the developed material was introduced in a package containing fresh prawns or chicken, its color was efficiently modified after one day of storage at ambient conditions demonstrating the possibility to be used as a food spoilage indicator. We prove by CIELab analysis that all color changes can be easily perceived visually, as the differences between the initial and the final color of the indicator after its interaction with the modified environment are well above the limit for visual perception. Therefore, the presented colorimetric porous indicator is suitable for multiple applications dealing with environmental protection and intelligent food packaging.

- Light Responsive Silk Nanofibers: An Optochemical Platform for Environmental Applications
  ACS Applied Materials & Interfaces 2017 9 (46), 40707-40715
  by M.E. Genovese, G. Caputo, G. Nanni, C. Setti, M. Bustreo, G. Perotto, A. Athanassiou, D. Fragouli

  Abstract: Photochromic spiropyran-doped silk fibroin poly(ethylene oxide) nanofibers which combine the attractive properties and biocompatibility of silk with the photocontrollable and reversible optical,

mechanical, and chemical response of the spiropyran dopants are herein presented. As proved, the reversible variation of the absorption and emission signals of the mats and of their Young's modulus upon alternate UV and visible light irradiation is ascribed to the reversible photoconversion of the spiropyran form to its polar merocyanine counterpart. Most importantly, the interactions of the merocyanine molecules with acidic vapors as well as with heavy metal ions dispersed in solution produce analyte-specific spectral changes in the emission profile of the composite, accompanied by a characteristic chromic variation. Because of the high surface-to-volume ratio of the nanofibrous network, such interactions are fast, thus enabling both an optical and a visual detection in a 30–60 s time scale. The sensing platform can be easily regenerated for more than 20 and 3 cycles upon acid or ion depletion, respectively. Overall, the photocontrolled properties of the silk composites combined with a straightforward preparation method render them suitable as porous materials and scaffolds with tunable compliance and reusable nanoprobes for real time optical detection in biomedical, environmental, and industrial applications.

Conference Papers

- Enhancing Visual Embeddings through Weakly Supervised Captioning for Zero-Shot Learning

  IEEE International Conference on Computer Vision Workshops (ICCVw), 1st International Workshop on Multi-Discipline Approach for Learning Concepts - Zero-Shot, One-Shot, Few-Shot and Beyond, 2019

  by M. Bustreo, J. Cavazza, V. Murino

  Abstract: Visual features designed for image classification have shown to be useful in zero-shot learning (ZSL) when generalizing towards classes not seen during training. In this paper, we argue that a more effective way of building visual features for ZSL is to extract them through captioning, in order not just to classify an image but, instead, to describe it. However, modern captioning models rely on a massive level of supervision, *e.g.*up to 15 extended descriptions per instance provided by humans, which is simply not available for ZSL benchmarks. In the latter in fact, the available annotations inform about the presence/absence of attributes within a fixed list only. Worse, attributes are seldom annotated at the image level, but rather, at the class level only: because of this, the annotation cannot be visually grounded. In this paper, we deal with such a weakly supervised regime to train an end-to-end LSTM captioner, whose backbone CNN image encoder can provide better features for ZSL. Our enhancement of visual features, called "*VisEn*", is compatible with any generic ZSL method, without requiring changes in its pipeline (apart from adapting hyper-parameters). Experimentally, *VisEn* is capable of sharply improving recognition performance on unseen classes, as we demonstrate through an ablation study which encompasses different ZSL approaches. Further, on the challenging fine-grained CUB dataset, *VisEn* improves by margin state-of-the-art methods, by using visual descriptors of one order of magnitude smaller.

- Analysis of Face-Touching Behavior in Large Scale Social Interaction Dataset

ACM International Conference on Multimodal Interaction (ICMI 2020)
by C. Beyan, M. Bustreo, M. Shahid, G. L. Bailo, N. Carissimi, A. Del Bue

Abstract: We present the first publicly available annotations for the analysis of face-touching behavior. These annotations are for a dataset composed of audio-visual recordings of small group social interactions with a total number of 64 videos, each one lasting between 12 to 30 minutes and showing a single person while participating to four-people meetings. They were performed by in total 16 annotators with an almost perfect agreement (Cohen's Kappa=0.89) on average. In total, 74K and 2M video frames were labelled as *face-touch* and *no-face-touch*, respectively.

Given the dataset and the collected annotations, we also present an extensive evaluation of several methods: rule-based, supervised learning with hand-crafted features and feature learning and inference with a Convolutional Neural Network (CNN) for Face-Touching detection. Our evaluation indicates that among all, CNN performed the best, reaching 83.76% F1-score and 0.84 Matthews Correlation Coefficient.

To foster future research in this problem, code and dataset were made publicly available (github.com/IIT-PAVIS/Face-Touching-Behavior), providing all video frames, face-touch annotations, body pose estimations including face and hands key-points detection, face bounding boxes as well as the baseline methods implemented and the cross-validation splits used for training and evaluating our models.

- Weakly Supervised Geodesic Segmentation of Egyptian Mummy CT Scans

International Conference on Pattern Recognition (ICPR 2020)
by A. Hati, M. Bustreo, D. Sona, V. Murino, A. Del Bue

Abstract: In this paper, we tackle the task of automatically analyzing 3D volumetric scans obtained from computed tomography (CT) devices. In particular, we address a particular task for which data is very limited: the segmentation of ancient Egyptian mummies CT scans. We aim at digitally unwrapping the mummy and identify different segments such as body, bandages and jewelry. The problem is complex because of the lack of annotated data for the different semantic regions to segment, thus discouraging the use of strongly supervised approaches. We, therefore, propose a weakly supervised and efficient interactive segmentation method to solve this challenging problem. After segmenting the wrapped mummy from its exterior region using histogram analysis and template matching, we first design a voxel distance measure to find an approximate solution for the body and bandage segments. Here, we use geodesic distances since voxel features as well as spatial relationship among voxels is incorporated in this measure. Next, we refine the solution using a GrabCut based segmentation together with a tracking method on the slices of the scan that assigns labels to different regions in the volume,

using limited supervision in the form of scribbles drawn by the user. The efficiency of the proposed method is demonstrated using visualizations and validated through quantitative measures and qualitative unwrapping of the mummy.

- A Versatile Crack Inspection Portable System based on Classifier Ensemble and Controlled Illumination

International Conference on Pattern Recognition (ICPR 2020)

by M. Gajanan Padalkar, C. Beltran-Gonzalez, M. Bustreo, A. Del Bue, V. Murino

Abstract: This paper presents a novel setup for automatic visual inspection of cracks in ceramic tile as well as studies the effect of various classifiers and height-varying illumination conditions for this task. The intuition behind this setup is that cracks can be better visualized under specific lighting conditions than others. Our setup, which is designed for field work with constraints in its maximum dimensions, can acquire images for crack detection with multiple lighting conditions using the illumination sources placed at multiple heights. Crack detection is then performed by classifying patches extracted from the acquired images in a sliding window fashion. We study the effect of lights placed at various heights by training classifiers both on customized as well as state-of-the-art architectures and evaluate their performance both at patch-level and image-level, demonstrating the effectiveness of our setup. More importantly, ours is the first study that demonstrates how height-varying illumination conditions can affect crack detection with the use of existing state-of-the-art classifiers. We provide an insight into the illumination conditions that can help in improving crack detection in a challenging real-world industrial environment.

- Complex-Object Visual Inspection: Empirical Studies on A Multiple Lighting Solution

International Conference on Pattern Recognition (ICPR 2020)

by M. Aghaei, M. Bustreo, P. Morerio, N. Carissimi, A. Del Bue, V. Murino

Abstract: The design of an automatic visual inspection system is usually performed in two stages. While the first stage consists in selecting the most suitable hardware setup for highlighting most effectively the defects on the surface to be inspected, the second stage concerns the development of algorithmic solutions to exploit the potentials offered by the collected data.

In this paper, first, we present a novel illumination setup embedding four illumination configurations to resemble diffused, dark-field, and front lighting techniques. Second, we analyze the contributions brought by deploying the proposed setup in the training phase only, mimicking the scenario in which an already developed visual inspection system cannot be modified on the customer site. Along with an exhaustive set of experiments, in this paper, we demonstrate the suitability of the proposed setup for effective illumination of complex-objects, defined as manufactured items with variable surface characteristics that cannot be determined a priori. Eventually, we provide insights into the importance of

multiple light configurations availability during training and their natural boosting effect which, without the need to modify the system design in the evaluation phase, lead to improvements in the overall system performance.

- Single Image Human Proxemics Estimation for Visual Social Distancing
  Winter Conference on Applications of Computer Vision (WACV 2020)
  by M. Aghaei, M. Bustreo, Y. Wang, G. L. Bailo, P. Morerio, A. Del Bue

  Abstract: In this work, we address the problem of estimating the so-called "Social Distancing" given a single uncalibrated image in unconstrained scenarios. Our approach proposes a semi-automatic solution to approximate the homography matrix between the scene ground and image plane. With the estimated homography, we then leverage an off-the-shelf pose detector to detect body poses on the image and to reason upon their inter-personal distances using the length of their body-parts. Inter-personal distances are further locally inspected to detect possible violations of the social distancing rules. We validate our proposed method quantitatively and qualitatively against baselines on public domain datasets for which we provided groundtruth on inter-personal distances. Besides, we demonstrate the application of our method deployed in a real testing scenario where statistics on the inter-personal distances are currently used to improve the safety in a critical environment.

Under Evaluation

- End-to-end pairwise human proxemics from uncalibrated single images
  Conference on Computer Vision and Pattern Recognition (CVPR 2021)
  by P. Morerio, M. Bustreo, Y. Wang, A. Del Bue

  Abstract: In this work, we address the ill-posed problem of estimating pairwise metric distances between people using only a single uncalibrated image. Our method *DeepProx* uses an end-to-end model that accepts as input two skeletal joints as a set of 2D image coordinates and it outputs the metric distance between them. We show that an increased performance is given by a geometrical loss over simplified camera parameters provided at training time. Further, *DeepProx* achieves a remarkable generalisation over novel viewpoints through domain generalisation. We validate our proposed method quantitatively and qualitatively against baselines on public datasets for which we provided groundtruth on interpersonal distances. Besides, we demonstrate the application of *DeepProx* deployed in a real testing scenario where pairwise proxemics are currently used to provide statistics on social distancing and thus improving the safety in critical environments.

References

[1] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[2] M. Marvin and A. P. Seymour, "Perceptrons," 1969.

[3] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, p. 574, 1959.

[4] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, vol. 1, no. 2, pp. 119–130, 1988.

[5] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

[6] P. Werbos, "Beyond regression: new tools for prediction and analysis in the behavioral sciences," *Ph. D. dissertation, Harvard University*, 1974.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[8] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, dec 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[16] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137, 2015.

[17] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[18] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in neural information processing systems*, pp. 892–900, 2016.

[19] G. Marcus, "Deep Learning: A Critical Appraisal," *arXiv preprint arXiv:1801.00631*, jan 2018.

[20] H. W. Lin, M. Tegmark, and D. Rolnick, "Why does deep and cheap learning work so well?," *Journal of Statistical Physics*, vol. 168, no. 6, pp. 1223–1247, 2017.

[21] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *arXiv preprint arXiv:1708.08296*, 2017.

[22] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

[23] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on fairness, accountability and transparency*, pp. 77–91, 2018.

[24] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[25] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.

[26] Y. Xian, *Learning from Limited Labeled Data - Zero-Shot and Few-Shot Learning*. PhD thesis, Faculty of Mathematics and Computer Science of Saarland University, 2020.

[27] M. J. H. Mohajeri and P. J. Manning, "Aria: an operating system of pavement distress diagnosis by image processing," *Transportation Research Record*, no. 1311, 1991.

[28] R. S. Walker and R. L. Harris, "Noncontact pavement crack detection system," *Transportation Research Record*, vol. 1311, pp. 149–157, 1991.

[29] A. Ammouche, J. Riss, D. Breysse, and J. Marchand, "Image analysis for the automated study of microcracks in concrete," *Cement and concrete composites*, vol. 23, no. 2-3, pp. 267–278, 2001.

[30] S. Iyer and S. K. Sinha, "A robust approach for automatic detection and segmentation of cracks in underground pipeline images," *Image and Vision Computing*, vol. 23, no. 10, pp. 921–933, 2005.

[31] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: Automatic crack detection from pavement images," *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, 2012.

[32] G. Medioni, C.-K. Tang, and M.-S. Lee, "Tensor voting: Theory and applications," in *Proceedings of RFIA*, vol. 2000, 2000.

[33] F.-C. Chen and M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and Naive Bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, 2017.

[34] S. Park, S. Bang, H. Kim, and H. Kim, "Patch-based crack detection in black box images using convolutional neural networks," *Journal of Computing in Civil Engineering*, vol. 33, no. 3, p. 4019017, 2019.

[35] L. Yang, B. Li, W. Li, B. Jiang, and J. Xiao, "Semantic metric 3d reconstruction for concrete inspection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1543–1551, 2018.

[36] X. Dong, C. J. Taylor, and T. F. Cootes, "Small defect detection using convolutional neural network features and random forests," in *Proceedings of the European Conference on Computer Vision (ECCV)*, p. 0, 2018.

[37] L. Yang, B. Li, G. Yang, Y. Chang, Z. Liu, B. Jiang, and J. Xiao, "Deep Neural Network based Visual Inspection with 3D Metric Measurement of Concrete Defects using Wall-climbing Robot.," in *IROS*, pp. 2849–2854, 2019.

[38] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1525–1535, 2019.

[39] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[40] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, pp. 234–241, 2015.

[41] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.

[42] B. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, pp. 442–451, oct 1975.

[43] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE transactions on image processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[44] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE, 2005.

[45] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, sep 2014.

[46] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[47] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, pp. 155–161, 1997.

[48] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1, pp. 278–282, IEEE, 1995.

[49] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, jun 2009.

[51] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, pp. 3856–3866, 2017.

[52] J. E. See, C. G. Drury, A. Speed, A. Williams, and N. Khalandi, "The role of visual inspection in the 21st century," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 61, pp. 262–266, SAGE Publications Sage CA: Los Angeles, CA, 2017.

[53] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, apr 2018.

[54] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European conference on computer vision*, pp. 740–755, 2014.

[55] K. Boyd, V. S. Costa, J. Davis, and D. Page, "Unachievable Region in Precision-Recall Space and Its Effect on Empirical Evaluation," *Proceedings of the International Conference on Machine Learning*, p. 349, jun 2012.

[56] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.

[57] R. Salakhutdinov, A. Torralba, and J. Tenenbaum, "Learning to share visual appearance for multiclass object detection," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2011.

[58] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly," *transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, 2018.

[59] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks.," in *Conference on Artificial Intelligence (AAAI)*, AAAI, 2008.

[60] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2009.

[61] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *International Conference on Computer Vision (ICCV)*, IEEE, 2015.

[62] V. K. Verma and P. Rai, "A simple exponential family framework for zero-shot learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, Springer, 2017.

[63] Y. Annadani and S. Biswas, "Preserving semantic relations for zero-shot learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2018.

[64] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016.

[65] H. Jiang, R. Wang, S. Shan, Y. Yang, and X. Chen, "Learning discriminative latent attributes for zero-shot classification," in *International Conference on Computer Vision (ICCV)*, IEEE, 2017.

[66] H. Jiang, R. Wang, S. Shan, and X. Chen, "Learning class prototypes via structure alignment for zero-shot recognition," in *European Conference on Computer Vision (ECCV)*, Springer, 2018.

[67] J. Song, C. Shen, J. Lei, A.-X. Zeng, K. Ou, D. Tao, and M. Song, "Selective Zero-Shot Classification with Augmented Attributes," in *European Conference on Computer Vision (ECCV)*, Springer, 2018.

[68] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean, "Zero-shot learning by convex combination of semantic embeddings," in *International Conference on Learning Representation (ICLR)*, 2014.

[69] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[70] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.

[71] A. Roy, J. Cavazza, and V. Murino, "Visually-Driven Semantic Augmentation for Zero-Shot Learning," in *British Machine Vision Conference (BMVC)*, BMVA, 2018.

[72] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning Deep Representations of Fine-Grained Visual Descriptions," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016.

[73] L. Anne Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell, "Deep compositional captioning: Describing novel object categories without training data," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016.

[74] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Guided open vocabulary image captioning with constrained beam search," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[75] Y. Wu, L. Zhu, L. Jiang, and Y. Yang, "Decoupled novel object captioner," in *2018 ACM Multimedia Conference on Multimedia Conference*, ACM, 2018.

[76] J. Lu, J. Yang, D. Batra, and D. Parikh, "Neural baby talk," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2018.

[77] X. Wang, J. Wu, D. Zhang, Y. Su, and W. Y. Wang, "Learning to Compose Topic-Aware Mixture of Experts for Zero-Shot Video Captioning," *Conference on Artificial Intelligence (AAAI)*, 2018.

[78] B. Wang, L. Ma, W. Zhang, and W. Liu, "Reconstruction network for video captioning," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2018.

[79] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," tech. rep., California Institute of Technology, 2011.

[80] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 36, no. 3, pp. 453–465, 2014.

[81] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 38, no. 7, pp. 1425–1438, 2016.

[82] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, and Others, "Devise: A deep visual-semantic embedding model," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[83] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015.

[84] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele, "Latent embeddings for zero-shot classification," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016.

[85] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning (ICML)*, 2015.

[86] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *International Conference on Computer Vision (ICCV)*, IEEE, 2015.

[87] M. Noroozi, H. Pirsiavash, and P. Favaro, "Representation learning by learning to count," in *International Conference on Computer Vision (ICCV)*, IEEE, 2017.

[88] S. Jenni and P. Favaro, "Self-supervised feature learning by learning to spot artifacts," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2018.

[89] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning-the good, the bad and the ugly," in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.

[90] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research (JMLR)*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[91] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research (JMLR)*, vol. 2, no. Dec, pp. 265–292, 2001.

[92] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Classifier and Exemplar Synthesis for Zero-Shot Learning," *International Journal of Computer Vision*, 2018.

[93] L. Anne Hendricks, R. Hu, T. Darrell, and Z. Akata, "Grounding visual explanations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 264–279, 2018.

[94] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "f-VAEGAN-D2: A Feature Generating Framework for Any-Shot Learning," *arXiv preprint arXiv:1903.10132*, 2019.

[95] M. Bustreo, J. Cavazza, and V. Murino, "Enhancing Visual Embeddings through Weakly Supervised Captioning for Zero-Shot Learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019.

[96] Y. L. Kwok, J. Gralton, and M. L. McLaws, "Face touching: a frequent habit that has implications for hand hygiene," *American Journal of Infection Control*, vol. 43, pp. 112–114, 2015.

[97] M. Nicas and D. Best, "A Study Quantifying the Hand-to-Face Contact Rate and Its Potential Application to Predicting Respiratory Tract Infection," *Journal of Occupational and Environmental Hygiene*, vol. 5, no. 6, pp. 347–352, 2008.

[98] M. Mahmoud and P. Robinson, "Interpreting Hand-Over-Face Gestures," in *Affective Computing and Intelligent Interaction*, pp. 248–255, 2011.

[99] M. Mahmoud, T. Baltrušaitis, and P. Robinson, "Automatic Analysis of Naturalistic Hand-Over-Face Gestures," *ACM Trans. Interact. Intell. Syst.*, vol. 6, no. 2, 2016.

[100] B. Nojavanasghari, C. E. Hughes, T. Baltrusaitis, and L.-P. Morency, "Hand2Face: Automatic synthesis and recognition of hand over face occlusions," in *Seventh International Conference on Affective Computing and Intelligent Interaction, ACII 2017, San Antonio, TX, USA, October 23-26, 2017*, pp. 209–215, IEEE Computer Society, 2017.

[101] S. Dimond and R. Harries, "Face touching in monkeys, apes and man: Evolutionary origins and cerebral asymmetry," *Neuropsychologia*, vol. 22, no. 2, pp. 227–233, 1984.

[102] T. Hatta and S. J. Dimond, "Differences in face touching by Japanese and British people," *Neuropsychologia*, vol. 22, no. 4, pp. 531–534, 1984.

[103] A. Macias Hernández, A. la Torre, S. Moreno Espinosa, P. E. Leal, M. Bourlon, and G. Ruiz-Palacios, "Controlling the novel A (H1N1) influenza virus: don't touch your face!," *The Journal of hospital infection*, vol. 73, pp. 280–281, 2009.

[104] P. Lunn, C. Belton, F. McGowan, S. Timmons, and D. Robertson, "Using Behavioural Science to Help Fight the Coronavirus: A Rapid, Narrative Review," *Journal of Behavioral Public Administration*, vol. 3, p. 2020, 2020.

[105] S. Michie, M. M. van Stralen, and R. West, "The behaviour change wheel: A new method for characterising and designing behaviour change interventions," *Implementation Science*, vol. 6, 2011.

[106] "Behavioural strategies for reducing covid-19 transmission in the general population." URL: https://implementationscience.biomedcentral.com/articles/10.1186/1748-5908-6-42#citeas.

[107] A. Zunino, J. Cavazza, R. Volpi, P. Morerio, A. Cavallo, C. Becchio, and V. Murino, "Predicting Intentions from Motion: The Subject-Adversarial Adaptation Approach," *International Journal of Computer Vision*, vol. 128, no. 1, pp. 220–239, 2020.

[108] I. Hasan, F. Setti, T. Tsesmelis, V. Belagiannis, S. Amin, A. Del Bue, M. Cristani, and F. Galasso, "Forecasting People Trajectories and Head Poses by Jointly Reasoning on Tracklets and Vislets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2019.

[109] E. Vats and C. S. Chan, "Early detection of human actions—A hybrid approach," *Applied Soft Computing*, vol. 46, pp. 953–966, 2016.

[110] X. Wang, J.-F. Hu, J.-H. Lai, J. Zhang, and W.-S. Zheng, "Progressive Teacher-Student Learning for Early Action Prediction," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[111] M. Mahmoud, T. Baltrušaitis, P. Robinson, and L. Riek, "3D corpus of spontaneous complex mental states," in *Conference on Affective Computing and Intelligent Interaction*, 2011.

[112] C. Beyan, N. Carissimi, F. Capozzi, S. Vascon, M. Bustreo, A. Pierro, C. Becchio, and V. Murino, "Detecting Emergent Leader in a Meeting Environment Using Nonverbal Visual Features Only," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pp. 317–324, 2016.

[113] K. Morita, K. Hashimoto, M. Ogata, H. Tsutsumi, S.-i. Tanabe, and S. Hori, "Measurement of Face-touching Frequency in a Simulated Train," *E3S Web of Conferences*, vol. 111, p. 2027, 2019.

[114] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust Face Landmark Estimation under Occlusion," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pp. 1513–1520, IEEE Computer Society, 2013.

[115] W. Yue and J. Qiang, "Robust Facial Landmark Detection under Significant Head Poses and Occlusion," in *Proceedings of the 2016 IEEE International Conference on Computer Vision*, ICCV '16, IEEE Computer Society, 2016.

[116] S. Saito, T. Li, and H. Li, "Real-Time Facial Segmentation and Performance Capture from RGB Input," in *Proceedings of the 14th European Conference on Computer Vision and Pattern Recognition, (ECCV 2016)*, pp. 244–261, Springer International Publishing, 2016.

[117] A. Behera, P. Matthew, A. Keidel, P. Vangorp, H. Fang, and S. Canning, "Associating Facial Expressions and Upper-Body Gestures with Learning Tasks for Enhancing Intelligent Tutoring Systems," *International Journal of Artificial Intelligence in Education*, 2020.

[118] C. Beyan, F. Capozzi, C. Becchio, and V. Murino, "Prediction of the Leadership Style of an Emergent Leader Using Audio and Visual Nonverbal Features," *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 441–456, 2018.

[119] "Computer Vision Annotation Tool: A Universal Approach to Data Annotation." URL: https://software.intel.com/en-us/articles/computer-vision-annotation-tool-a-universal-approach-to-data-annotation.

[120] "Powerful and efficient Computer Vision Annotation Tool (CVAT)." URL: https://github.com/opencv/cvat.

[121] M. L. McHugh, "Interrater reliability: the kappa statistic," in *Biochemia medica*, 2012.

[122] C. Beyan and R. Fisher, "Classifying imbalanced data sets using similarity based hierarchical decomposition," *Pattern Recognition*, vol. 48, no. 5, pp. 1653–1672, 2015.

[123] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[124] B. Xiao, P. G. Georgiou, B. Baucom, and S. S. Narayanan, "Data driven modeling of head motion towards analysis of behaviors in couple interactions," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3766–3770, 2013.

[125] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, 2009.

[126] C. Busso, Z. Deng, M. Grimm, U. Neumann, and S. Narayanan, "Rigid Head Motion in Expressive Speech Animation: Analysis and Synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1075–1086, 2007.

[127] C. Beyan, A. Zunino, M. Shahid, and V. Murino, "Personality Traits Classification Using Deep Visual Activity-based Nonverbal Features of Key-Dynamic Images," *IEEE Transactions on Affective Computing*, 2019.

[128] S. Okada, L. S. Nguyen, O. Aran, and D. Gatica-Perez, "Modeling Dyadic and Group Impressions with Intermodal and Interperson Features," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 1s, 2019.

[129] D. Jayagopi, H. Hung, C. Yeo, and D. Gatica-Perez, "Modeling dominance in group conversations using nonverbal activity cues," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 3, pp. 501–513, 2009.

[130] D. Sanchez-Cortes, O. Aran, M. S. Mast, and D. Gatica-Perez, "A Nonverbal Behavior Approach to Identify Emergent Leaders in Small Groups," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 816–832, 2012.

[131] S. K. Brooks, R. K. Webster, L. E. Smith, L. Woodland, S. Wessely, N. Greenberg, and G. J. Rubin, "The psychological impact of quarantine and how to reduce it: rapid review of the evidence," *The Lancet*, 2020.

[132] J. Johnson Jr, S. Hasan, D. Lee, C. Hluchan, and N. Ahmed, "Social-Distancing Monitoring Using Portable Electronic Devices," tech. rep., Technical Disclosure Commons, 2020.

[133] M. Cristani, A. D. Bue, V. Murino, F. Setti, and A. Vinciarelli, "The Visual Social Distancing Problem," *IEEE Access*, vol. 8, pp. 126876–126886, 2020.

[134] A. Criminisi, I. Reid, and A. Zisserman, "Single View Metrology," *International Journal of Computer Vision*, vol. 40, pp. 123–148, 2000.

[135] M. Aghaei, M. Bustreo, Y. Wang, P. Morerio, and A. Del Bue, "Single Image Human Proxemics Estimation for Visual Social Distancing," in *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, 2021.

[136] L. Bertoni, S. Kreiss, and A. Alahi, "Monoloco: Monocular 3d pedestrian localization and uncertainty estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6861–6871, 2019.

[137] Y. Chen, Y. Tian, and M. He, "Monocular human pose estimation: A survey of deep learning-based methods," *Computer Vision and Image Understanding*, vol. 192, p. 102897, 2020.

[138] H. Hung and B. Kröse, "Detecting F-formations as Dominant Sets," in *International Conference on Multimodal Interfaces (ICMI)*, pp. 231–238, 2011.

[139] Y. Yang, S. Baker, A. Kannan, and D. Ramanan, "Recognizing proxemics in personal photos," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3522–3529, 2012.

[140] M. Cristani, L. Bazzani, G. Paggetti, A. Fossati, D. Tosato, A. Del Bue, G. Menegaz, and V. Murino, "Social interaction discovery by statistical analysis of F-formations," in *British Machine Vision Conference (BMVC)*, pp. 23.1—-23.12, 2011.

[141] I. Chakraborty, H. Cheng, and O. Javed, "3D Visual Proxemics: Recognizing Human Interactions in 3D from a Single Image," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3406–3413, 2013.

[142] G. Rogez, P. Weinzaepfel, and C. Schmid, "Lcr-net++: Multi-person 2d and 3d pose detection in natural images," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[143] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-Shot Multi-person 3D Pose Estimation from Monocular RGB," in *Proceedings of International Conference on 3D Vision (3DV)*, pp. 120–130, 2018.

[144] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "Xnect: Real-time multi-person 3d human pose estimation with a single rgb camera," *arXiv preprint arXiv:1907.00837*, 2019.

[145] J. Li, C. Wang, W. Liu, C. Qian, and C. Lu, "HMOR: Hierarchical Multi-Person Ordinal Relations for Monocular Multi-Person 3D Pose Estimation," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[146] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint Triplets for Object Detection," in *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6568–6577, 2019.

[147] M. Rezaei and M. Azarmi, "DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic," *Applied Sciences*, vol. 10, no. 21, p. 7514, 2020.

[148] M. Fabbri, F. Lanzi, R. Gasparini, S. Calderara, L. Baraldi, and R. Cucchiara, "InterHomines: Distance-Based Risk Estimation for Human Safety," 2020.

[149] M. Fabbri, F. Lanzi, S. Calderara, S. Alletto, and R. Cucchiara, "Compressed Volumetric Heatmaps for Multi-Person 3D Pose Estimation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[150] J. Y. Zhang, S. Pepose, H. Joo, D. Ramanan, J. Malik, and A. Kanazawa, "Perceiving 3D Human-Object Spatial Arrangements from a Single Image in the Wild," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[151] L. Bertoni, S. Kreiss, and A. Alahi, "Perceiving Humans: from Monocular 3D Localization to Social Distancing," 2020.

[152] D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 3–15, 2008.

[153] Y. Ganin and V. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1180–1189, PMLR, 2015.

[154] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-Adversarial Training of Neural Networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[155] K. Chaudhury, S. DiVerdi, and S. Ioffe, "Auto-rectification of user photos," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 3479–3483, IEEE, 2014.

[156] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera People Tracking with a Probabilistic Occupancy Map," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 267–282, 2008.

[157] T. Chavdarova, P. Baqué, A. Maksai, S. Bouquet, C. Jose, L. Lettry, F. Fleuret, P. Fua, and L. V. Gool, "WILDTRACK: A Multi-camera HD Dataset for Dense Unscripted Pedestrian Detection," in *Proc. of IEEE/CVF Conference On Computer Vision And Pattern Recognition*, (New York), pp. 5030–5039, IEEE, 2018.

[158] B. Benfold and I. Reid, "Stable Multi-Target Tracking in Real-Time Surveillance Video," in *CVPR*, pp. 3457–3464, 2011.

[159] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[160] C. Riggs, *Unwrapping Ancient Egypt*. Bloomsbury Publishing, 2014.

[161] F. Casali, "X-ray and neutron digital radiography and computed tomography for cultural heritage," *Physical Techniques in the Study of Art, Archaeology and Cultural Heritage*, vol. 1, pp. 41–123, 2006.

[162] S. Hughes, "CT scanning in archaeology," *Computed tomography-specialist applications*, vol. 2011, pp. 57–70, 2011.

[163] F. Cesarani, M. C. Martina, R. Boano, R. Grilletto, E. D'Amicone, C. Venturi, and G. Gandini, "Scenes from the Past. Multidetector CT study of gallbladder stones in a wrapped Egyptian mummy," *RadioGraphics*, vol. 29, no. 4, pp. 1191–1194, 2009.

[164] S. Curto and M. Mancini, "News of Kha and Meryt," *The Journal of Egyptian Archaeology*, vol. 54, no. 1, pp. 77–81, 1968.

[165] "Mummy Museo Egizio collection: N. inv. Suppl. 8431; XVIII Dinastia, regno di Amenhotep III, 1388-1351 a.C.."

[166] G. Wang, M. A. Zuluaga, W. Li, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, and T. Vercauteren, "DeepIGeoS: A deep interactive geodesic framework for medical image segmentation," *IEEE Trans. PAMI*, vol. 41, no. 7, pp. 1559–1572, 2018.

[167] N. Sharma and L. M. Aggarwal, "Automated medical image segmentation techniques," *Journal of Medical Physics*, vol. 35, no. 1, p. 3, 2010.

[168] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans. PAMI*, vol. 39, pp. 640–651, apr 2017.

[169] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. IEEE CVPR*, pp. 2814–2821, 2014.

[170] F. Zhao and X. Xie, "An overview of interactive medical image segmentation," *Annals of the BMVA*, vol. 2013, no. 7, pp. 1–22, 2013.

[171] F. Albertin, A. Patera, I. Jerjen, S. Hartmann, E. Peccenini, F. Kaplan, M. Stampanoni, and Others, "Virtual reading of a large ancient handwritten science book," *Microchemical Journal*, vol. 125, pp. 185–189, 2016.

[172] D. Stromer, V. Christlein, C. Martindale, P. Zippert, E. Haltenberger, T. Hausotte, and A. Maier, "Browsing through sealed historical manuscripts by using 3-D computed tomography with low-brilliance X-ray sources," *Scientific Reports*, vol. 8, no. 1, pp. 1–10, 2018.

[173] W. B. Seales, C. S. Parker, M. Segal, E. Tov, P. Shor, and Y. Porath, "From damage to discovery via virtual unwrapping: Reading the scroll from En-Gedi," *Science Advances*, vol. 2, no. 9, p. e1601247, 2016.

[174] R. Brancaccio, M. Bettuzzi, F. Casali, M. P. Morigi, G. Levi, A. Gallo, G. Marchetti, and D. Schneberk, "Real-Time Reconstruction for 3-D CT Applied to Large Objects of Cultural Heritage," *IEEE Transactions on Nuclear Science*, vol. 58, no. 4, pp. 1864–1871, 2011.

[175] A. Re, F. Albertin, C. Avataneo, R. Brancaccio, J. Corsi, G. Cotto, S. De Blasi, G. Dughera, E. Durisi, W. Ferrarese, and Others, "X-ray tomography of large wooden artworks: the case study of "Doppio corpo" by Pietro Piffetti," *Heritage Science*, vol. 2, no. 1, pp. 1–9, 2014.

[176] A. Huppertz, D. Wildung, B. J. Kemp, T. Nentwig, P. Asbach, F. M. Rasche, and B. Hamm, "Nondestructive insights into composition of the sculpture of Egyptian Queen Nefertiti with CT," *Radiology*, vol. 251, no. 1, pp. 233–240, 2009.

[177] M. C. Martina, F. Cesarani, R. Boano, A. M. Donadoni Roveri, A. Ferraris, and Others, "Kha and Merit: multidetector computed tomography and 3D reconstructions of two mummies from the Egyptian Museum of Turin," in *World Congress on Mummy Studies*, vol. 80, pp. 42–44, 2005.

[178] R. Bianucci, M. E. Habicht, S. Buckley, J. Fletcher, R. Seiler, L. M. Öhrström, E. Vassilika, T. Böni, and F. J. Rühli, "Shedding New light on the 18th dynasty mummies of the royal architect Kha and his spouse Merit," *PLoS One*, vol. 10, no. 7, p. e0131916, 2015.

[179] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *International Conference on 3D Vision (3DV)*, pp. 565–571, 2016.

[180] K. Kamnitsas, C. Ledig, V. F. J. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Med. Image Anal.*, vol. 36, pp. 61–78, 2017.

[181] J. Borovec, J. Švihlík, J. Kybic, and D. Habart, "Supervised and unsupervised segmentation using superpixels, model estimation, and graph cut," *Journal of Electron. Imaging*, vol. 26, no. 6, pp. 1–17, 2017.

[182] S. A. Haider, M. J. Shafiee, A. Chung, F. Khalvati, A. Oikonomou, A. Wong, and M. A. Haider, "Single-click, semi-automatic lung nodule contouring using hierarchical conditional random fields," in *International Symposium on Biomedical Imaging*, pp. 1139–1142, 2015.

[183] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 359–369, 1998.

[184] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[185] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in *Proc. ICCV*, pp. 1–8, 2007.

[186] M. Rajchl, M. C. H. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, W. Bai, M. Damodaram, M. A. Rutherford, and Others, "DeepCut: Object segmentation from bounding box annotations using convolutional neural networks," *IEEE Trans. Med. Imaging*, vol. 36, no. 2, pp. 674–683, 2016.

[187] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.

[188] J. E. Cates, A. E. Lefohn, and R. T. Whitaker, "GIST: An interactive, GPU-based level set segmentation tool for 3D medical images," *Med. Image Anal.*, vol. 8, no. 3, pp. 217–231, 2004.

[189] G. Wang, M. A. Zuluaga, R. Pratt, M. Aertsen, T. Doel, M. Klusmann, A. L. David, J. Deprest, and Others, "Slic-Seg: A minimally interactive segmentation of the placenta from sparse and motion-corrupted fetal MRI in multiple views," *Med. Image Anal.*, vol. 34, pp. 137–147, 2016.

[190] F. L. Bookstein, "Principal warps: thin-plate splines and the decomposition of deformations," *IEEE Trans. PAMI*, vol. 11, no. 6, pp. 567–585, 1989.

[191] P. Dollár, "Piotr's computer vision matlab toolbox (PMT)," 2014.

[192] B. A. Griffin and J. J. Corso, "Tukey-Inspired Video Object Segmentation," in *Proc. IEEE WACV*, 2019.