

CONTINUAL LEARNING OF PREDICTIVE MODELS IN VIDEO SEQUENCES VIA VARIATIONAL AUTOENCODERS

Damian Campo, Giulia Slavic, Mohamad Baydoun, Lucio Marcenaro, Carlo Regazzoni

DITEN, University of Genova, Italy

ABSTRACT

This paper proposes a method for performing continual learning of predictive models that facilitate the inference of future frames in video sequences. For a first given experience, an initial Variational Autoencoder, together with a set of fully connected neural networks are utilized to respectively learn the appearance of video frames and their dynamics at the latent space level. By employing an adapted Markov Jump Particle Filter, the proposed method recognizes new situations and integrates them as predictive models avoiding catastrophic forgetting of previously learned tasks. For evaluating the proposed method, this article uses video sequences from a vehicle that performs different tasks in a controlled environment.

Index Terms— Continual learning, lifelong learning, variational autoencoder, particle filter, kalman filter

1. INTRODUCTION

Some biological organisms, such as pigeons and large primates, possess the ability to learn new experiences continuously through their lifetime [1, 2, 3]. Such a capacity of preserving and use information from past experiences allows organisms to develop cognitive skills that are often critical for survival [4, 5]. As discussed in [6], the development of biological expertise follows a characteristic pattern of gradual improvement of performance over a particular task. The work in [6] also claims that the level of expertise reached by an organism depends on its *i)* long-term memory, *ii)* working memory capacity, *iii)* ability to focus attention on relevant information, *iv)* capability to anticipate, perceive and comprehend surroundings, *v)* velocity at the decision-making and *vi)* coordination in motor movements. The capabilities mentioned above can be seen as a set of cognitive abilities that organisms employ to solve problems and adapt to new situations systematically. We argue that each of those skills can be improved/refined by recalling past experiences that match with current situations, suggesting a major role of continual learning when solving problems and developing expertise.

Motivated by the aforementioned research studies on the continual learning in living beings, we propose a method by which video sequences acquired by artificial systems are employed to create models that can predict future situations based on past experiences. The proposed method facilitates the continual learning of new experiences by using abnormal information (video-frames) detected from available predictive models. We believe artificial systems can be highly benefited by the continual learning of new experiences, contributing to the automatic development of cognitive skills that facilitate the growth of expertise and adaptability in machines.

Several articles have tried to include continual learning (also known as lifelong learning) capabilities into artificial systems inspired by findings in psychology and neuroscience [7, 8]. Primarily, continual learning has been studied in deep neural networks (DNNs) due to their remarkable advances across diverse applications [9]. Nonetheless, when trying to integrate new information to artificial

neural networks (ANNs), it is observed a dramatic performance degradation in previously learned tasks, such a phenomenon is known as *catastrophic forgetting*; and various articles have tried to overcome it by using different techniques [10, 11, 12]. In particular, the work in [7] distinguishes three main approaches to deal with catastrophic forgetting, facilitating the continual learning in ANNs: *i) Architectural*, where the architecture of the network is modified, and the objective function remains the same. *ii) Functional*, which modifies the loss function, encouraging that the learning of new tasks does not affect already learned information. *iii) Structural*, consisting of introducing penalties on parameters to avoid forgetting already learned experiences.

The proposed method consists of an architectural approach that enables the continual learning of predictive models in video sequences by adopting a duplication and tuning process over Variational Autoencoders (VAEs), which are trained as new situations are detected. For a given task, our work uses a VAE for learning the appearance features of video data and a set of fully connected ANNs for performing predictions of following video frames at the latent space level. Our work demonstrates how predictive models can be learned and used incrementally as new situations are observed.

Although various research works have tackled the problem of continual learning in DNNs, their contributions have been mainly focused on classifications tasks [13, 14, 15] consisting in continuously learning different classes from diverse datasets such as CIFAR-10/100, ImageNet, and MNIST. On the other hand, our work focuses on the ability to predict the subsequent frames of video sequences by associating visual observations with already identified/learned experiences. Moreover, the proposed method employs a Markov Jump Particle Filter (MJPF) [16] over latent space information for making predictions that can be potentially integrated with other sensory information, e.g., positional and control data.

The main contributions of the proposed method are: *i)* The employment of VAEs that facilitate to obtain latent spaces from which to create predictive models at the low-dimensional level. *ii)* The detection of new situations that autonomous systems may employ to learn continuously predictive models without forgetting previously learned experiences. *iii)* For evaluation purposes, this paper uses real video sequences coming from a vehicle performing different tasks in a controlled environment.

The rest of the paper is organized as follows: Section 2 explains the proposed method for enabling continual learning over video sequences based on VAEs' latent spaces. Section 3 introduces the employed dataset. Section 4 discusses the obtained results and section 5 concludes the article and suggests future developments.

2. METHOD

The proposed method includes two major phases that are continuously repeated each time new situations are detected: *i)* a training process (section 2.1) to learn probabilistic models based on observed data, and an online testing procedure (section 2.2) for detecting pos-

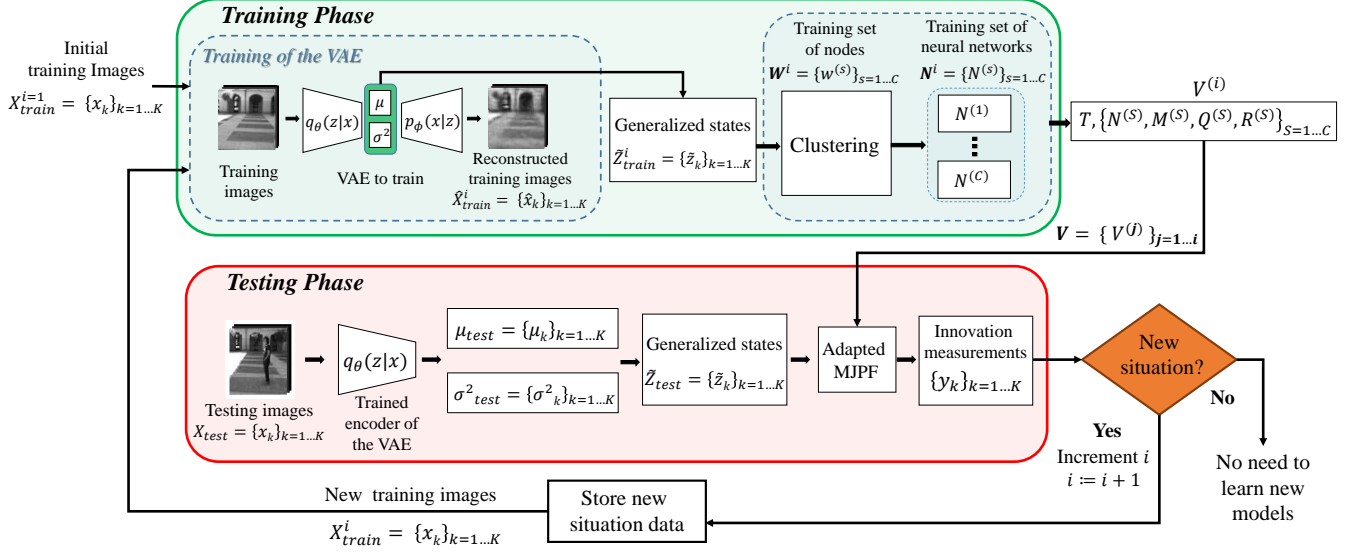


Fig. 1: Block diagram of proposed method.

sible new situations on observed data. Accordingly, when new situations are identified, they are employed to trigger a new training process that refines available predictive models (section 2.3). The proposed method is summarized in the block diagram shown in Fig. 1.

2.1. Training phase

Variational Autoencoder. A Variational Autoencoder (VAE) is used for describing images in a latent space that significantly reduces the original dimension of video frames. Moreover, a VAE facilitates to represent images in the latent state probabilistically by using a mean μ and variance σ^2 to approximate each latent variable. As is well known, a VAE is composed of two parts: an encoder $q_\theta(z|x)$ and a decoder $p_\phi(x|z)$. The latent state z sampled from $\mathcal{N}(\mu, \sigma^2)$, returns an approximate reconstruction of the observation x . Through θ and ϕ , we define the parameters of both encoder and decoder, respectively. To optimize them, the VAE maximizes the sum of the lower bound on the marginal likelihood of each observation x of the dataset D , as described in [17, 18].

This work uses the VAE's ability to encode visual information into a significant lower-dimensional probabilistic latent space, which is employed to make inferences of future instances. Consequently, we first train the VAE with a set of training images X_{train}^i , where i indexes the task to be learned (initially, $i = 1$). By utilizing the trained VAE's encoder, we obtain a set of latent features described by μ_{train} and σ_{train}^2 , which represent X_{train}^i data.

Generalized states. Let μ_{train} be a set training images' states corresponding to the VAE's latent space data; we build a set of Generalized States (GSs) containing also the first-order time derivatives of μ_{train} . Accordingly, let $\mu_k \in \mu_{train}$ be the state of the training image at time k , its first-order time derivative can be approximated by $\dot{\mu}_k \sim \frac{\mu_k - \mu_{k-1}}{\Delta k}$, $\dot{\mu}_k \in \dot{\mu}_{train}$. $\Delta k = 1$ assumes a normalized regular sampling of images. The GS at time k can thus be written as $\tilde{z}_k = [\mu_k \ \dot{\mu}_k]^T$. By repeating this for each image $x_k \in X_{train}^i$, we obtain a set of GSs for the training set, defined by:

$$\tilde{Z}_{train}^i = [\mu_{train} \ \dot{\mu}_{train}]^T. \quad (1)$$

Clustering and neural networks. After obtaining \tilde{Z}_{train}^i , we use a traditional k-means algorithm to cluster GSs into groups that carry similar information. Since we use μ and $\dot{\mu}$ as input data, obtained clusters capture information of encoded images and their dynamics.

By letting C be the total number of identified clusters, it is possible to use S to index clusters, such that $S \in \{1, \dots, C\}$. Once the clustering is performed, we calculate a transition matrix T encoding the passage probabilities from each cluster to the others. Consequently, the following features are extracted from each cluster S : *i*) cluster's centroid $M^{(S)}$, *ii*) cluster's covariance $Q^{(S)}$ and *iii*) cluster's radius of acceptance $R^{(S)}$. Finally, a fully connected neural network $N^{(S)}$ defining the dynamics of GSs, i.e., continuous predictive model, is learned for each cluster. For training each $N^{(S)}$, the value of every μ_k is taken as input and the corresponding $\dot{\mu}_{k+1}$ as output, where $[\mu_k, \dot{\mu}_k]^T \in S$, such that:

$$\dot{\mu}_{k+1} \sim N^{(S)}(\mu_k) + w_k, \quad (2)$$

where w_k is the residual error after the convergence of the network.

Each $N^{(S)}$ learns a sort of *quasi-semantic* information based on a particular image appearance and motion detected by the cluster S , facilitating the estimation of future latent spaces, i.e., predicting the following frames. Such predictions can be employed to measure the similarity between new observations and previously learned experiences encoded into NNs. In case predictions from NNs are not compliant with observations, an abnormality should be detected, and models should be adapted to learn new situations and semantic information. Consequently, each identified task i can be described by the set of parameters $V^{(i)} = \{T, \{N^{(S)}, M^{(S)}, Q^{(S)}, R^{(S)}\}_{S=1, \dots, C}\}$.

2.2. Testing Phase

During the testing phase, each image X_{test} is processed through the VAE, and their respective GSs are calculated. Then, an adapted version of the MJPF based on the learned information $V^{(i=1)}$ is used to detect new situations in video sequences.

Adapted Markov Jump Particle Filter. An MJPF, firstly proposed in [16], is adapted for prediction and abnormality detection purposes on visual data. The MJPF uses a Particle Filter coupled with a bank of Kalman Filters (KFs) for inferring continuous and discrete level information. Since this work tackles a problem that requires a non-linear predictive model and a non-linear observation model, solved respectively by the set of NNs and a VAE, it is employed a bank of unscented KFs (UKF) and VAE's encoded information for making inferences over video sequences.

The proposed adapted MJPF (A-MJPF) follows two stages at each time instant k : *prediction* and *update*. During prediction, the next cluster S_{k+1} (discrete level) and GS \tilde{z}_{k+1} (continuous level) are estimated for each particle, i.e., $p(S_{k+1}|S_k)$ and $p(\tilde{z}_{k+1}|\tilde{z}_k)$ respectively. The prediction at discrete level is similar to the standard MPJF in [16]. Instead, the A-MJPF uses the neural network $N^{(S_k)}$ to make predictions at a continuous level. Since non-linear models are considered for predicting continuous level information, a UKF is utilized as described in [19] by taking $2L$ additional sigma points. Each sigma point's prediction follows the equation below:

$$\tilde{z}_{k+1} = f(\tilde{z}_k) = A\tilde{z}_k + BN^{(S)}(\mu_k) + w_k, \quad (3)$$

A and B are matrices that map the previous state \tilde{z}_k^i and the predicted velocity computed by $N^{(S)}(\mu_k^i)$ on the new state \tilde{z}_{k+1}^i . $A = [A_1 A_2]$ with $A_1 = [I_L 0_{L,L}]^T$, $A_2 = 0_{2L,L}$; and $B = [I_L I_L]^T$. The mean and covariance of \tilde{z}_{k+1}^i are calculated through the UKF.

The update phase is performed when a new measurement (image) is observed. At the discrete level, particles are resampled based on an innovation measurement. At the continuous state level, a modified KF is in charge of the update. This update takes into consideration the fact that μ_k and σ_k^2 associated with x_k can be used as the mapped observation on the state space at time k . Consistently, σ_k^2 can approximate the covariance matrix, such that $\Sigma_k \sim I_L \sigma_k^2$, representing the uncertainty while encoding images. By assuming a negligible observation noise, it is possible to employ a modified version of the KF update equations where the observation matrix H disappears. Algorithm 1 describes the employed KF's steps.

Algorithm 1 Equations for the prediction and update phases of the Adapted Kalman Filter.

▷ **PREDICTION:**

- 1: Calculation of the sigma points $\tilde{z}_{k|k}^i$ and of their respective weights $\tilde{W}^{i,m}$ and $\tilde{W}^{i,c}$ as described in [19].
- 2: $\tilde{z}_{k+1|k}^i = f(\tilde{z}_{k|k}^i)$
- 3: $\tilde{z}_{k+1|k} = \sum_{i=0}^{2L} \tilde{W}^{i,m} \tilde{z}_{k+1|k}^i$
- 4: $P_{k+1|k} = \sum_{i=0}^{2L} \tilde{W}^{i,c} \{\tilde{z}_{k+1|k}^i - \tilde{z}_{k+1|k}\} \{\tilde{z}_{k+1|k}^i - \tilde{z}_{k+1|k}\}^T$
- 5: $P_{k+1|k}^L = P_{k+1|k} \Big|_{\{\text{row}:1\dots L, \text{col}:1\dots L\}}$

▷ **UPDATE:**

- 6: $K_{k+1} = [P_{k+1|k}^L; I_L] (P_{k+1|k}^L + \Sigma_{k+1})^{-1}$
- 7: $\tilde{z}_{k+1|k+1} = \tilde{z}_{k+1|k} + K_{k+1}(\mu_{k+1} - \mu_{k+1|k})$
- 8: $P_{k+1|k+1} = P_{k+1|k} - K_{k+1}(P_{k+1|k}^L + \Sigma_{k+1})K_{k+1}^T$

Detection of new situation. After the update phase, at each time instant k , the predicted value of $z_k^{l,p}$ related to latent state component l and particle p is compared with the actual updated value, outputting a measure of innovation defined as:

$$y_k = \min_p \frac{\sum_{l=1}^L |\mu_{k|k}^{l,p} - \mu_{k|k-1}^{l,p}|}{L}. \quad (4)$$

The innovation values of training video sequences are used to set a threshold defined as:

$$thresh = \bar{y}_{train} + 3std(y_{train}), \quad (5)$$

where \bar{y}_{train} and $std(y_{train})$ are the mean and standard deviation of innovations from the training data respectively. When applying algorithm 1 on testing data, frames producing innovation values above the threshold in Eq.(5) are considered as new situations. Moreover, to avoid spurious innovation peaks, a temporal window of 3 frames is used, such that new situations are recognized only if 3 consecutive frames are above *thresh*.

2.3. Continual Learning

The calculation of innovations facilitates defining a continual learning process where frames belonging to new situations are detected and stored. These identified frames are employed to perform a new training process as described in section 2.1, involving a new VAE ($VAE^{i:=i+1}$). μ_{train} and σ_{train}^2 are employed as bottleneck features related to the images of a new situation $X_{train}^{i:=i+1}$.

During the new testing phase, the outputted bottleneck features and learned feature variables V^j of the different VAEs VAE^j where $j = \{1, \dots, i+1\}$, are used together in a single A-MJPF. The particles in the A-MJPF are then distributed among all the available clusters and consequently among the various VAEs, i.e., situations.

Since the bottleneck features among VAEs capture a different meaning, in the MJPF, particles assigned to a particular VAE's cluster cannot be reassigned to other VAEs' clusters. Therefore, some particles do not jump between VAEs, but they always remain attached to a particular VAE. The innovation measurement (see Eq. 4) is again estimated in order to detect additional new situations.

3. EMPLOYED DATASET

A real vehicle called "iCab" [20], is used to collect video sequences from an onboard front camera. A human drives the iCab performing different tasks in a closed environment.

This work aims at studying situations that have not been previously seen in a normal situation (Scenario I), which is used for learning purposes. Scenario II includes unseen maneuvers caused by the presence of pedestrians while the vehicle performs a previously seen task. The two scenarios considered in this work are:

Scenario I (perimeter monitoring). The vehicle follows a rectangular trajectory around a closed building.

Scenario II (pedestrian avoidance maneuver). Two obstacles (stationary pedestrians) in different locations interfere with the perimeter monitoring task of Scenario I. The vehicle performs an avoidance maneuver and continues the perimeter monitoring. Fig. 3 shows a temporal evolution of video frames from both scenarios.

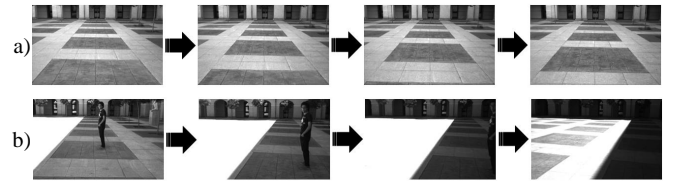


Fig. 3: Video sequences from considered scenarios: a) parameter monitoring task and b) pedestrian avoidance maneuver.

4. EXPERIMENTAL RESULTS

First training phase: Perimeter Monitoring. Initial models are trained based on the video frames from Scenario I. This data corresponds to $X_{train}^{i=1}$ and facilitates the obtainment of $VAE^{(1)}$ and the set of features in $V^{(1)}$, see section 2.1. The threshold in Eq.(5) is then calculated based on the behavior of the training data on the initial A-MJPF. We call $MODEL^{(PM)}$ the model using $VAE^{(1)}$ and the A-MJPF based on $V^{(1)}$.

Detection of new situation: Pedestrian Avoidance. The icab faces a new situation: it encounters and avoids a static pedestrian. Fig. 2a) shows the resulting innovation signal from $MODEL^{(PM)}$; blue zones refer to video frames containing the pedestrian and yellow regions encode the avoidance maneuvers. At each lap, the vehicle encounters two different static pedestrians, see images (1) and (6) in Fig. 2a). They wear t-shirts of different colors (black and white), which make them "camouflage" with the environment in

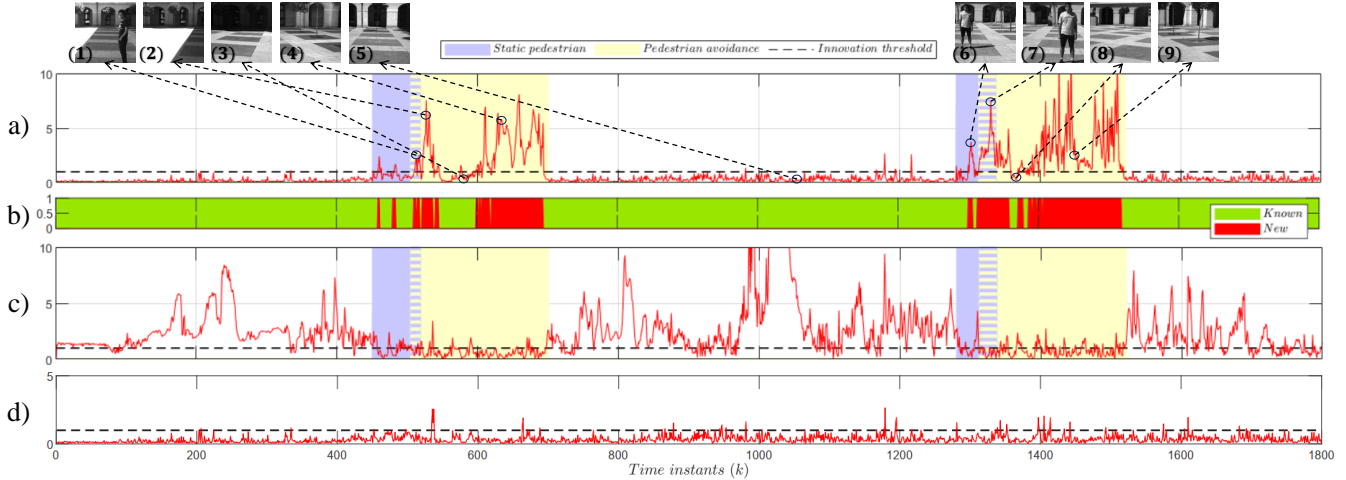


Fig. 2: Testing phases on the pedestrian avoidance task: (a) Innovation signal using $MODEL^{(PM)}$. (b) Color-coded innovation. (c) Innovation signal using $MODEL^{(PA)}$. (d) Innovation signal using $MODEL^{(PM,PA)}$.

some particular configurations due to changeable illumination conditions. This factor influences the innovation values of frames (1) and (6), with the second one generating a higher values.

Each maneuver of pedestrian avoidance generates two peaked zones, see frames(2) and (4) or (7) and (9). Between such peaks, there is a zone with low innovation values, see (3) or (8), due to the execution of similar behaviors already observed in the training set.

As described in section 2.2, the amplitude threshold obtained from the initial experience and a temporal window of 3 frames are used to detect the new situations. Fig. 2b) displays the frames that were classified as known experiences (green) or new situations (red).

4.1. Learning of the new situation

The frames classified as new are used as $X_{train}^{i=2}$ for generating an additional VAE ($VAE^{(2)}$) and set of feature variables $V^{(2)}$ which can be used for generating a model that understands only the pedestrian avoidance maneuver; we call it $MODEL^{(PA)}$. Innovation measurements from $MODEL^{(PA)}$ are displayed in Fig. 2c) where low innovations are obtained in zones related to the pedestrian presence and vehicle’s avoidance maneuver.

Innovation signals from $MODEL^{(PM)}$ and $MODEL^{(PA)}$ can be seen as complementary information that facilitates the incremental understanding of the proposed two tasks together, see how large innovations values in Fig. 2a) correspond to low innovations in c). Accordingly, by employing available VAEs and variables: $VAE^{(1)}$, $VAE^{(2)}$, $V^{(1)}$, and $V^{(2)}$, we generate a single A-MPJF, called $MODEL^{(PM,PA)}$ that uses all previously learned concepts for prediction purposes, see section 2.3. Fig.2d) shows how innovations from $MODEL^{(PM,PA)}$ remain low through the entire scenario II, confirming the continual learning of new experiences avoiding the catastrophic forgetting of previously learned concepts. The percentage of false positive alarms of $MODEL^{(PM,PA)}$ is 7.14%.

Fig. 4 displays three examples that explain the performance of our algorithm visually. From left to right the columns of the blue block correspond to: the image at time k ; its reconstruction by using respectively $VAE^{(1)}$ and $VAE^{(2)}$; the decoded version of the predicted image at time $k + 1$ given the image at time k when adopting respectively $MODEL^{(PM)}$, $MODEL^{(PA)}$ and $MODEL^{(PM,PA)}$. Similarly, columns of the green block represent: the image at $k + 1$; its reconstruction when using respectively $VAE^{(1)}$ and $VAE^{(2)}$.

The row a) corresponds to the perimeter monitoring task, whereas rows b) and c) are related to the pedestrian avoidance sit-

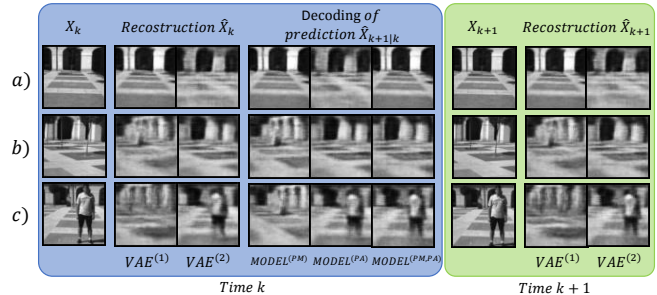


Fig. 4: Examples of image-level behaviour in perimeter monitoring (a) and pedestrian avoidance (b-c) cases.

uation. X_{k+1} and innovation values correspond to the ones shown in Fig. 2a) with a blue circle, see frames (5), (7), (9). Note how in a), the prediction of $MODEL^{(PM)}$ is accurate, while the one of $MODEL^{(PA)}$ is not, due to the wrong reconstruction of the image. In cases b) and c), the prediction of $MODEL^{(PA)}$ is accurate and the prediction of $MODEL^{(PM)}$ is not. This lousy performance of $MODEL^{(PM)}$ in case c) is again due to the image not being recognized, leading to inconsistencies while predicting. In the case of b), $VAE^{(1)}$ can generalize the observed image to a similar one that was in the training set. However, the prediction still produces high innovations because of a discrepancy between the expected motion and the observed dynamics: instead of moving left, the vehicle moves right to finalize the pedestrian avoidance maneuver. It can be visually observed how the prediction of $MODEL^{(PM,PA)}$ performs well in all three cases.

5. CONCLUSION AND FUTURE WORK

The proposed work proposes a method that facilitates the continual learning of dynamical situations in video data. The proposed method is based on a probabilistic approach that uses latent spaces from VAEs to represent the state of video frames at each time instant. The dynamics of video sequences are captured by a set of NNs that encode different types of video motions in a given task. Future work includes the insertion of multimodal data into the A-MPJF, allowing the model to make inferences by fusing heterogeneous sensory data, e.g., video and positional information. Another possible path of the proposed work consists of improving the clustering process of latent space information, such that richer semantics can be obtained.

6. REFERENCES

- [1] Joël Fagot and Robert G Cook, “Evidence for large long-term memory capacities in baboons and pigeons and its implications for learning and the evolution of cognition,” *Proceedings of the National Academy of Sciences*, vol. 103, no. 46, pp. 17564–17567, 2006.
- [2] Murray Shanahan, Verner P Bingman, Toru Shimizu, Martin Wild, and Onur Güntürkün, “Large-scale network organization in the avian forebrain: a connectivity matrix and theoretical analysis,” *Frontiers in computational neuroscience*, vol. 7, pp. 89, 2013.
- [3] Alice E Milne, Christopher I Petkov, and Benjamin Wilson, “Auditory and visual sequence learning in humans and monkeys using an artificial grammar learning paradigm,” *Neuroscience*, vol. 389, pp. 104–117, 2018.
- [4] Michael J Beran, Charles R Menzel, Audrey E Parrish, Bonnie M Perdue, Ken Sayers, J David Smith, and David A Washburn, “Primate cognition: attention, episodic memory, prospective memory, self-control, and metacognition as examples of cognitive control in nonhuman primates,” *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 7, no. 5, pp. 294–316, 2016.
- [5] Kevin P Darby, Leyre Castro, Edward A Wasserman, and Vladimir M Sloutsky, “Cognitive flexibility and memory in pigeons, human children, and adults,” *Cognition*, vol. 177, pp. 30–40, 2018.
- [6] Reuven Dukas, “Animal expertise: mechanisms, ecology and evolution,” *Animal behaviour*, vol. 147, pp. 199–210, 2019.
- [7] Friedemann Zenke, Ben Poole, and Surya Ganguli, “Continual learning through synaptic intelligence,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3987–3995.
- [8] Timo Flesch, Jan Balaguer, Ronald Dekker, Hamed Nili, and Christopher Summerfield, “Comparing continual task learning in minds and machines,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 44, pp. E10313–E10322, 2018.
- [9] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter, “Continual lifelong learning with neural networks: A review,” *Neural Networks*, 2019.
- [10] Zhizhong Li and Derek Hoiem, “Learning without forgetting,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.
- [11] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al., “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.
- [12] Xin Yao, Tianchi Huang, Chenglei Wu, Rui-Xiao Zhang, and Lifeng Sun, “Adversarial feature alignment: Avoid catastrophic forgetting in incremental task lifelong learning,” *Neural computation*, vol. 31, no. 11, pp. 2266–2291, 2019.
- [13] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert, “icarl: Incremental classifier and representation learning,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.
- [14] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner, “Variational continual learning,” *arXiv preprint arXiv:1710.10628*, 2017.
- [15] Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Andrei Stoian, and David Filliat, “Generative models from the perspective of continual learning,” in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.
- [16] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, “Learning switching models for abnormality detection for autonomous driving,” in *2018 21st International Conference on Information Fusion (FUSION)*, July 2018, pp. 2606–2613.
- [17] Diederik P. Kingma and Max Welling, “Auto-encoding variational bayes.,” in *ICLR*, Yoshua Bengio and Yann LeCun, Eds., 2014.
- [18] Diederik P. Kingma and Max Welling, “An introduction to variational autoencoders.,” *Foundations and Trends in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.
- [19] E.A. Wan and R. van der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Symposium on Adaptive Systems for Signal Processing, Communication and Control*. 2000, IEEE.
- [20] Pablo Marín-Plaza, Jorge Beltrán, Ahmed Hussein, Basam Musleh, David Martín, Arturo de la Escalera, and José María Armingol, “Stereo vision-based local occupancy grid map for autonomous navigation in ros,” *Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, 2016.