# Video Action Classification Using Deep Predictive Coding Networks

Roshan Rane Vageesh Saxena Edit Szügyi

Advisor: André Ofner
Applications of Deep Learning Project Module
MSc Cognitive Systems
University of Potsdam



## Abstract

*In this paper, we evaluate the PredNet [13] on the Something-something action data set [14] and implement the PredNet+, which we train in a multi-task fashion to output both classification labels and predictions. Our idea is to condition video prediction and action classification on each other. We discuss a series of observations about the PredNet and conclude that it does not completely follow the principles of the predictive coding framework.*

## 1. Introduction

Understanding the content of visual data is an important prerequisite for successful computer vision applications. Until recently, computer vision models were almost exclusively trained on static images, achieving exceptional results [3] [20] [21]. However, in the real visual world, both the objects and the viewer are in constant movement and static images cannot completely represent that. In order to leverage this additional information, video analysis has become a prolific area of research in machine learning. Researchers have suggested that the movement of objects through time is an important signal for learning about the structure of the object [15] [19]. Due to the complexity of the task, unsurprisingly, different kinds of deep neural network architectures perform best [29].

Most research regarding video classification and captioning has been initiated by the emergence of large labelled data sets, such as the HMDB [12], the ActivityNet [10] or the Sports-1M [11]. Labelled data sets containing action videos are an especially important tool for teaching models about the real world. However, labelled data is expensive and time-consuming to get, and the more recent crowd-sourced data sets are of questionable quality. Unsupervised learning is therefore a promising direction. Video prediction [5] [15] [24] [25] [26] is one possible way of learning from unlabelled data. A key insight is that in order to be able to predict how the visual world will change over time, the model should learn about the object structure and the possible transformations an object can undergo [13]. The model should also learn to focus on regions that change between consequent frames, which is more sample efficient, as it enables the model to learn from far fewer data samples. Thus, we have chosen an architecture exclusively designed to perform unsupervised video prediction.

A recent trend in video prediction is based on the predictive coding theory from the neuroscience literature [16], which posits that the brain is continually making predictions of incoming sensory stimuli. Top-down connections convey these predictions, which are compared against actual observations to generate an error signal. This error signal gets propagated back up the hierarchy, eventually leading to an update of the predictions. In this paper, we examine and extend the PredNet, a deep neural network architecture, designed on the principles of predictive coding [13].

Our contribution to this growing body of research, towards better comprehension of videos are two-fold:

1. We evaluate the effectiveness of the PredNet [13] on a challenging and robust action classification data set, the Something-something [14].

2. We combine the PredNet's unsupervised video prediction with supervised video classification in a multi-task training fashion. Our novel idea is to condition future predictions on the predicted label classes and vice versa.

## 2. Related work

### 2.1. Action video classification

A crucial design decision in video analysis is how to handle the time component of the videos. Some widespread approaches are using deep spatial CNNs [11], 3D-CNNs [22], using two different streams for the spatial and the temporal components of the video [4] [18], or temporal structure

modelling [25]. See an extensive overview in [29].

Another important factor is the nature of the action data and the labels. Most existing large data sets have coarse-grained action labels. This means that the models are trained on a relatively easy task, which is getting a long sequence as an input and are expected to produce only one vector as an output. This action vector may be detected even from isolated frames, e.g. 'soccer' can be inferred from a soccer field. To overcome this issue and force the models to learn a better world model, Mahdisoltani et al. [14] work with their improved version of the Something-something data set [8], which contains 174 fine-grained action labels, as for instance 'putting something on a table', 'pretending to put something on a table', or 'putting something on a slanted surface so it slides down.' Their paper provides evidence for the hypothesis that task granularity is strongly correlated with the quality and generalizability of the learned features. As for the nature of the data, being crowd-sourced, it includes noise much resembling the real world: thousands of different objects, variations of lighting conditions, background patterns, and camera motion.

### 2.2. Video prediction

As already pointed out, training a model to predict future frames of a video offers many benefits. Finn et al. [5] model pixel-level motion by predicting a distribution over pixel motion from previous frames and conditioning on a robot's future actions. Srivastava et al. [19] use an LSTM-based model and demonstrate that predicting the future sequences of high-level representations of frames (percepts) in an unsupervised manner improves video classification results. Villegas et al. [23] use the raw frames as well as their high-level representation, which is the corresponding human pose in the frame, to predict long-term into the future. However, they require labelled pose information and work only with static backgrounds. Vondrick et al. [24] continue in this vein of encoding images at a higher-than-pixel level. They apply recognition algorithms on the predicted representation to anticipate objects and actions. Unlike the above methods, the PredNet model we explore [13], learns directly from the pixel-space, and works with videos with non-static background and real world settings. Furthermore, the model is based on a neuroscientific framework that is posited to learn features at different hierarchical levels without being exclusively tuned to do so.

### 2.3. Predictive coding in deep learning

Lotter et al. [13] introduce the PredNet, a network that learns to predict future frames, with each layer in the network making local predictions and only forwarding deviations from those predictions to subsequent network layers. They claim that the PredNet learns internal representations of the objects and can capture important features.

Their paper is a starting point for various researchers. For instance, Zhong et al.'s [30] AFA-PredNet further integrates the motor action as an additional signal which modulates the top-down generative process via an attention mechanism. Taking this one step further, they [31] propose a predictive hierarchical artificial model where different temporal scales of predictions exist on different levels of the hierarchical predictive coding, which are defined in the temporal parameters in the neurons. Wen et al. [28] use predictive coding in object recognition. They implement a network with feedforward connections carrying the prediction errors to higher-layers, feedback connections that carry the prediction to its lower-layers, and recurrent connections for local memory. Han et al. [9] build on this model and develop a bidirectional and dynamic neural network with local recurrent processing which they call predictive coding network (PCN).

## 3. Models

### 3.1. PredNet architecture

The PredNet architecture is shown in Figure 1 [13]. The network is made of stacked hierarchical layers, each of which attempts to make local predictions of its input. The difference between the actual input and this prediction is then passed up the hierarchy to the next layer. Information flows in three ways through the network: (1) the error signal flows in the bottom-up direction as marked by the red arrows on the right, (2) the prediction signal flows in the top-down direction as shown by the green arrow on the left, and (3) the local error signal and prediction estimation signal flow within each layer. Every layer consists of four units: an input convolution unit ($A_i$), a recurrent representation unit ($R_i$) followed by a prediction unit ($Ahat_i$) and an error calculation unit ($E_i$) as labelled in Figure 1. The representation unit is made of a ConvLSTM [17] layer that estimates what the input will be on the next time step. This is fed into the prediction unit, which is made of a convolution layer that generates the prediction $Ahat_i$. The error units calculate the difference between the prediction and the input. Moreover, it splits them into positive and negative error populations to add more non-linearity. This error is then fed as input to the next layer. The representation unit receives a copy of the error signal (red arrow) along with the up-sampled input from the representation unit of the higher-level (green arrow), which it uses along with its recurrent memory to perform future predictions.

### 3.2. PredNet+ Architecture

For the second phase of our project, we modify the PredNet architecture such that it performs label classification for the videos along with future frame predictions. We informally call this architecture PredNet+ and it is shown in Fig-

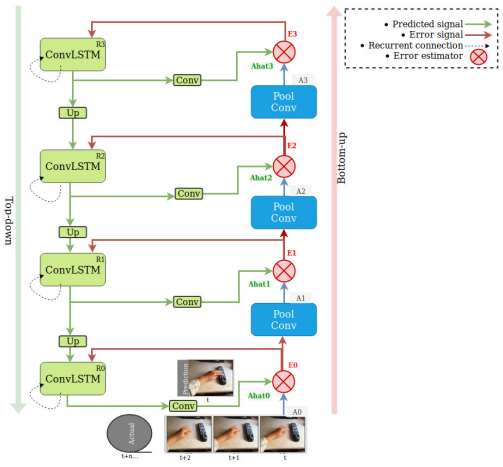


Figure 1. PredNet [13] architecture.

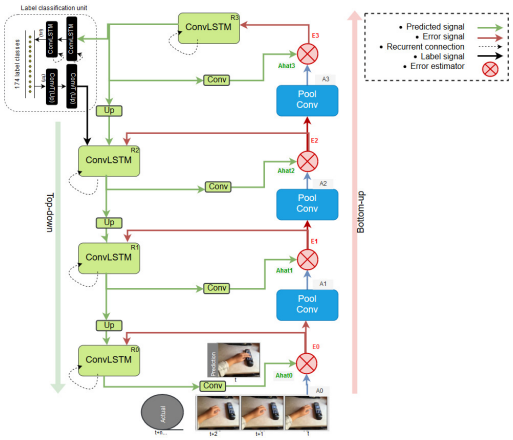


Figure 2. PredNet+ architecture.

ure 2. As seen in said figure, along with the vanilla PredNet units it contains an additional 'label classification unit' that is attached to the top-most representation layer. This unit in turn consists of an encoder section and a decoder section. In Figure 2 the two ConvLSTM layers (in black) form the encoder which transforms the output of the representation unit $R_3$ into label class probabilities. The two transposed convolution layers (also in black) make up the decoder that up-samples and transforms the label classes back to the image modality, which is further fed back into the top down as shown by the black arrow to $R_2$. The rationale for choosing this particular architecture is discussed further in section 5.1.

# 4. Probing the PredNet

## 4.1. Evaluation metrics

Defining a good evaluation metric for the quality of image predictions is a challenging subtopic in itself [2][15]. There is no universally accepted measurement of image quality and consequently none for image similarity either. Mean squared error (MSE) is a commonly used measure because it is computationally light. However, it often does not correlate with human perception, for instance, two images can have the exact same MSE while one is blurry and the other one is sharp. In comparison, the mean average error (MAE) is more robust to outliers, since it does not use the square. Another improved metric based on the MSE is the Peak Signal Noise Ratio (PSNR) [15]. With both MSE and PSNR, only deviations between corresponding pixels are considered and are treated as being independent of their neighboring pixels. The Structural Similarity Index Measure (SSIM) [27] counters this. It is a perception-based metric that considers image degradation as a perceived change in structural information, while also incorporating important perceptual phenomena.

Following Mathieu et al. [15], we use two metrics which calculate PSNR and SSIM only for the frames which have movement with respect to the previous frames. We call these 'PSNR movement' and 'SSIM movement' respectively. In our case this is crucial as action videos often include only few frames of movement. Rewarding a model for correctly predicting stillness would thus be counterproductive. Another variation of the SSIM we have experimented with, is what we informally name 'conditioned SSIM', which is calculated as seen in Equation 1. This metric quantifies how different the predictions are from the previous frame and therefore would give us a measure of how 'risky' predictions of our model are in comparison to just 'last-frame-copying'.

$$SSIM_{cond} = (SSIM_{max} - SSIM(actual_{t-1}, pred_t)) * SSIM(actual_t, pred_t) \quad (1)$$

Furthermore, we have experimented with two different kind of sharpness measures, one based on the gradient of the images, according to Mathieu et al. [15] and another one that is built on Fourier-transformations, described in Kanjar et al. [2]. Yet, the sharpness scores solely reflect the image size. If the original image has a low resolution, even if the predictions are of low quality, the difference between the original and the prediction will be small. One useful insight from this metric has been that the predictions of all our models have worse scores than the original video, pointing out that all of them add a lot of blurring, as shall be described in Section 4.2.

A graph with all the metrics described above, for all our models (listed in Table 1) is provided in Appendix A. After

careful manual comparison of each of these metric scores against actual prediction quality, we have decided that three metrics are most informative and can together reflect the prediction quality best: the PSNR movement, SSIM movement and the conditioned SSIM. We therefore use these metrics in the rest of our paper.

### 4.2. Experiments

In the first phase of our project, we evaluate the PredNet's performance on a rich action classification data set. We try out different visualization techniques and report the insights as well as various shortcomings of the PredNet. We experiment with 10 models with different combinations of frames-per-second (FPS), number of layers, image size, and the number of channels per layer. All our models are listed in Table 1 and their performances with different metrics are presented in Appendix A.

In order to better understand what each module at each layer of the PredNet does, we conducted extensive visualization experiments. See Figure 8 for one complete example and Appendix B for two more. The inspiration for the lower part of the graph came from the error maps in the paper by Han et al. [9]. The frame matrices are averages of all channels, so for instance in Layer 1, R0 the first frame would be an average of 32 different channels. We produced them for all models and a number of hand-selected videos but due to the size of the image, we have decided to only include one in the paper. In this section, we dedicate one paragraph to each of our findings, and Figure 8 will aid in discussing them.

**The top-down connections in the PredNet are reactive and not predictive.** This can be seen in the average R plots of the figure. The top-down learns to only react to large errors in the incoming frame, caused by movements and does not proactively predict them before the movements actually starts. In general, the learning dynamics of our best model is as follows: The model simply performs previous-frame-copy if there are no cues for motion in the previous two consecutive frames. If there is a cue for motion, there are two possibilities for the predicted frame: if the direction of the motion is predictable and smooth, it interpolates the object in the direction of the motion. If this is not the case, it smudges the region containing the object of motion to effectively reduce the loss.

**The PredNet model learns useful features only when trained on videos with continuous motion.** The PredNet was designed and tested by Lotter et al. [13] on videos which do not have a lot of still frames, such as their own synthetic 'Rotating Faces data set' and the KITTI data set [6]. The KITTI data set consists of videos captured by a camera roof-mounted on a car in an urban setting. The data is complex in the sense that it contains a variety of objects, perspectives and lighting conditions. However, there is always an uniform movement of both the camera and the objects on the scene. This is in stark contrast with our action data set which can have a lot of still frames, see e.g. Figure 3. In this scenario, the PredNet resolves to performing mere last-frame copying, as it is statistically beneficial to do just that. It performs 'interesting' predictions only after it encounters a large error signal because of movement in the video, which points back to the previous finding: the PredNet is reactive and not actually predictive. If the model is not motivated enough to make 'true' predictions rather than trivial last-frame-copying, then it does not need to learn the dynamics of how the objects and the scene moves and therefore the features it learns are not very useful. All of our further steps are means to improve the PredNet's performance on our action data set.


Figure 3. Example frames of low FPS input.

**The PredNet's learning ability is sensitive to frames-per-second (FPS) rate.** On the KITTI data, Lotter et al. [13] used a small number of frames at a fixed FPS rate. This is also different with action data. If we sample only a few frames from the video, we might miss the action in the video completely, whereas, working with a large number of frames is computationally expensive. One way to deal with this is to reduce the FPS rate. Lowering the FPS however, results in abrupt motion between frames. This creates an unrealistic expectation from the model to learn to predict objects suddenly appearing in the frame out of nowhere. See Figure 3 for an example. Another motivation for trying lower FPS rates is that it would force the model to learn 'risky' predictions. Yet, this only aggravates the problem as lowering the FPS does not reduce the proportion of the video's still frames. For instance, if the FPS rate is 12 and 20 out of 24 frames had no motion then changing the FPS rate to 3, 5 out of 6 frames would also not contain any motion. Hence, the proportion of still frames statistically remains the same and the model will continue to resort to last-frame-copying. After experimenting with FPS rates of 3, 6 and 12, we settle with 12. High FPS rate causes the input to have smooth transitions of objects entering and exiting the frames and thus, the model has a better chance of learning something 'useful'. See a comparison of performances of models with different FPS rates in Figure 4. Model 2 performs best according to the movement PSNR and the conditioned SSIM, while the SSIM improvement is slightly better for Model 1. We have manually checked the results and saw that Model 2 makes more 'interesting' predictions and thus have made our decision to settle on the highest FPS rate.

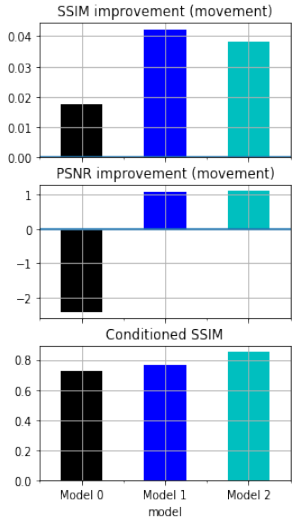


Figure 4. Results of Models with 3, 6 and 12 FPS rate.

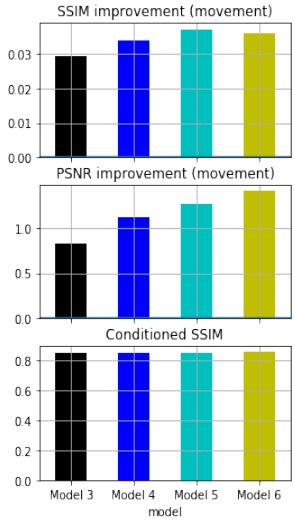


Figure 5. Results of Models with 4, 5, 6 and 7 layers.


Figure 6. Example frames of no multi-modality.

**The PredNet performs interesting predictions only when the higher layers have full receptive field.** After experimenting with image size and the number of layers, we discover that the receptive field is of utmost importance. We tried lower image resolutions to reduce the computation time and increase the receptive field of the convolution filters in the highest representation units. However, many videos contain lots of clustered objects and background and reducing the resolution too much makes them uninterpretable. Another way to enhance the receptive field is to have deeper layers. As we increase the layers, the results get iteratively better, see the graph in Figure 5. Yet, models with 7 layers and above start causing memory issues. Therefore, our best performing models: Model 3 and Model 7 (see Appendix A) are those which have full receptive field at the top-most representation layers.

**The representation units do not learn multi-modal distributions.** The representation units in PredNet are deterministic in the sense that they learn to perform the 'one best' prediction but do not learn the probability distribution of all possible future states. They especially do not have the capacity to learn multi-modal distributions, which is a common place in video prediction. For instance, see Figure 6. In this example there are two possible ways the thumb in the picture could move and hence there are two equally-probable future states. In this scenario, the model just resorts to regression-to-the-mean and just blurs the regions with movement or does last-frame-copy as seen in most predictions. This is further supported by the experiments we conducted with different sharpness measures. The predictions by the model are always less sharp than the actual videos.

**The PredNet is not a perfect emulation of the predictive coding framework.** It is evident from Figure 8, that the average error increases as we go up the layers, which is the exact opposite of what is expected by predictive coding: The errors should reduce as we go up as every layer tries to 'explain away' portions of the incoming error. While the principle idea of the PredNet is to combine the very successful convolutional network architecture from deep learning research with the predictive coding framework, it fails to really do so as it does not harness the depth of the network without deviating from the predictive coding principles. In Lotter et al. [13], they show that models trained with $L_0$ loss perform better than those trained with $L_{all}$ loss on the KITTI data. $L_0$ loss means that the model is trained to minimize only the error $E_0$ on the lowest layer (see Figure 1) and $L_{all}$ loss means that the model is trained to minimize the errors on all the layers. The Predictive Coding framework dictates $L_{all}$ loss. We retest this with the Something-something data set and get the same results as shown in Figure 7.

This, along with the visualizations, proves that the PredNet tries to leverage the depth in its architecture by **operating as two sub-modules** which is possible only with the $L_0$ loss. In the 2 sub-module setup, the lowest layer of the PredNet, $R_0$ works to generate the $(t+1)$ predictions of the incoming frames using the error $E_0$ at time $t$ and context $R_1$ from upper layers. Meanwhile, the rest of the layers operate as one 'deep' unit to produce the most useful transformation of $E_0$ and provides it to $R_0$ as the context $R_1$. In Figure 8 it

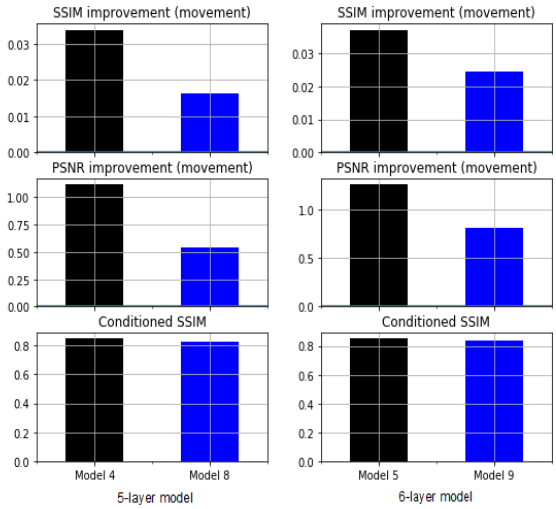


Figure 7. Results of Models with $L_0$ and $L_{all}$ loss.

can be noted that the average R value for Layer 1 is independent of the other three and the errors and representations of the first layer are distinct from the other layers. We can also see when looking at the frames that $R_0$ has a different behaviour from the higher $R_i$. Namely, $R_0$ is trying to recreate the exact frame, while the higher $R_i$ abstract from it. One possible fix is to make each representation layer deep by itself, for instance like the multi-scale net architecture in the paper by Mathieu et al. [15]. However, this would be very computationally expensive. The optimal solution, therefore, would be to design the network such that the 'hierarchical' nature of the network is exploited efficiently, which also facilitates a proper implementation of predictive coding with $L_{all}$ loss.

| **Model** | **fps** | **Layers** | **Image (h x w)** | **No. of param.** | **Loss** $L_{0/all}$ |
|---|---|---|---|---|---|
| 0 | **3** | 4 | 48 X 56 | 6.9 | $L_0$ |
| 1 | **6** | 4 | 48 X 56 | 6.9 | $L_0$ |
| 2 | **12** | 4 | 48 X 56 | 6.9 | $L_0$ |
| 3 | 12 | **4** | 32 X 48 | 6.9 | $L_0$ |
| 4 | 12 | **5** | 48 X 80 | 5.3 | $L_0$ |
| 5 | 12 | **6** | 64 X 96 | 5.8 | $L_0$ |
| 6 | 12 | **7** | 128 X 192 | 6.2 | $L_0$ |
| 7 | 12 | 6 | **96 X 160** | **7.2** | $L_0$ |
| 8 | 12 | 5 | 48 X 80 | 5.3 | $L_{all}$ |
| 9 | 12 | 6 | 64 X 96 | 5.8 | $L_{all}$ |

Table 1. Experiments with different frames-per-seconds (FPS), number of layers, no of parameters in the model (in millions), the image size and whether it was trained with $L_0$ loss or $L_{all}$ loss. Comparable models are grouped using horizontal lines and the column that varies are in bold.

### 4.3. Extrapolation

Following the work of Lotter et al. [13], we further evaluate the PredNet by testing its ability to extrapolate into the future. The framework for extrapolation is as follows: once extrapolation is started at time step $T$, the model's predictions are fed back into the network as the next input frame. This is done iteratively until the end of the video.

In our first experiment, we test the ability of our best model, Model 7, to extrapolate into the future without being exclusively trained to do so. (This is formally referred to as $(t+1)$ model in the Figures.) We experiment with three different starting points of extrapolation: at $t/4$, $t/2$, and at $3t/4$ time steps, where $t$ stands for the total number of time steps or frames in the whole video. The predictions for all the three different starting points are shown in Figure 9. As it can be seen, the model does not have enough contextual information to extrapolate the flow of motion into the future when starting at $t/4$ or $t/2$. This is in particular due to motion mostly starting at the middle or towards the end of the video in our data set. The metrics are evaluated only for the predictions made after the extrapolation starts. Based on this analysis, we decide on $3t/4$ as the best starting point for extrapolation.

In our second experiment, we have fine-tuned the best model for extrapolation at $3t/4$ time step by exclusively training it in the extrapolation framework for additional 30k videos. The final extrapolated predictions are given in Appendix C. In this example, we can faintly observe that the rotation of the object (the RAM chip) continues in clockwise direction for two more steps into the future after the start of the extrapolation. However, it is difficult to infer if the model has really learned the projection of the object's movement, because of the extensive blurring in the regions of motion.

As already pointed out in Section 4.2, the PredNet's architecture compels it to perform predictions in a reactive manner by using the movement between consecutive input frames as an active cue. Therefore, when we feed the predictions back as input, the model gets a cue that the action has stopped and reverts back to performing last-frame-copying. Eventually, the predictions get blurrier over time. This is because, the minor blur added by the down-sampling units in the bottom-up and up-sampling units in the top-down accumulate exponentially over time. Finally, this behaviour also suggests that the network is designed to only learn short-term interpolation and does not build long-term hypotheses. Nevertheless, we must keep in mind that the training procedure can be improved by testing the extrapolation at every time step $T$ of the video and thus, the results are not conclusive.

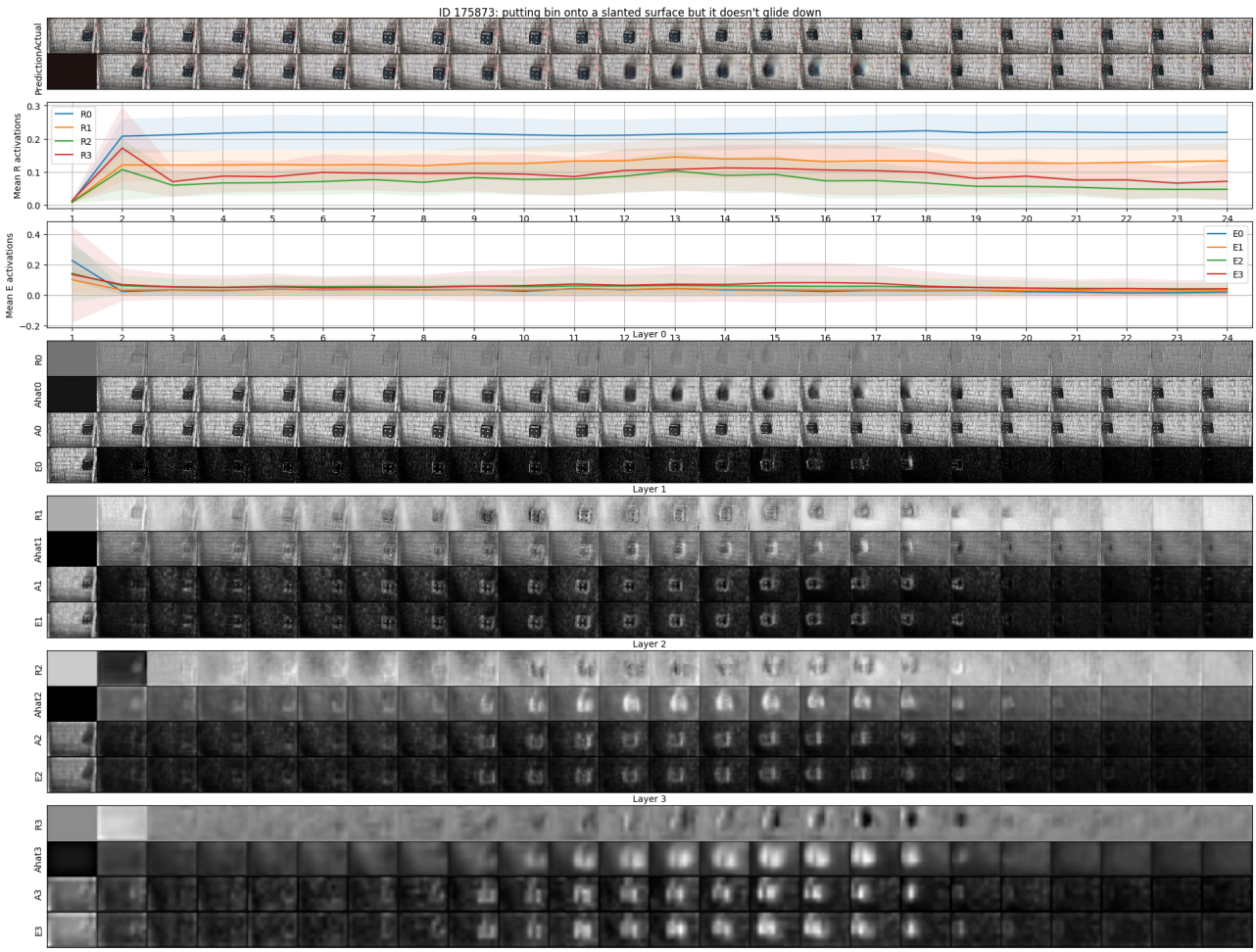


Figure 8. Model 7 full visualization.


Figure 9. Extrapolation results for Model 7 extrapolated at different time steps. The red mark shows the start of the extrapolation.

## 5. Label classification with PredNet+

The extensive experimentation we have performed in the first phase of this project has helped us to understand the inner workings of the PredNet architecture. In this section, we further test the architecture by modifying it to perform supervised label classification simultaneously with unsupervised video prediction. For a comparison of the architectures of PredNet+ and the vanilla PredNet, see Figures 2 and 1 respectively. The model design is described in section 3.2 and the rationale for the design and the results are discussed further below.

### 5.1. Design of PredNet+

The 'label prediction unit' makes predictions at each incoming frame. A weighted sum of these prediction scores is calculated and passed through a softmax function to get the final predicted class probabilities for the whole video. The model does not have enough context to make meaningful predictions in the first few frames of the video and there-

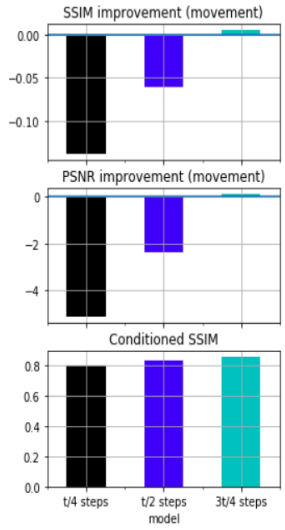


Figure 10. Evaluation of extrapolated model 7 at $t + n$ steps.

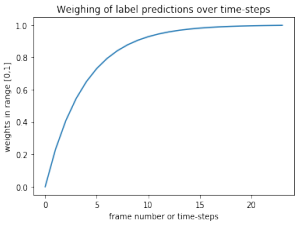


Figure 11. Weighing of label predictions over time-steps.

fore the weighing-over-time is done using an exponential function shown in Figure 11. Notice how the predictions in the first few frames are weighed low while the weights for later predictions stabilize at 1.0. The PredNet+ is designed such that it shares latent features at the top-most representation layer for its two tasks of label classification and future frames prediction. The future frame predictions are conditioned on the label predictions made by the 'label classification unit' (shown in Figure 2 by the black arrow going into $R_2$). We hypothesized that this would improve the results on both sub-tasks as evident in many multi-task training scenarios [1] [7]. Even though in our case, we attach the 'label classification unit' to the top-most layer, this is not necessarily the only approach nor the best one as per predictive coding. But as already discussed in Section 4.2, PredNet behaves more like a conventional deep neural network. Also, the top-most layer in our models are tuned to 'see' the whole video i.e. they have full receptive field on the frames and contain recurrent memory. For these reasons and for ease of implementation, we have decided to plug in the 'label classification unit' at the top-most layer.

In summary, the 'label classification unit' and the PredNet units in PredNet+, are expected to work in tandem in a multitask learning set-up and form a synergy. Yet, this is not what we observe in our results.

| Model | Top-1 | Top-5 |
|---|---|---|
| Baseline [8] | 11.5 | 30.0 |
| Ours | 28.2 | 57.0 |
| Mahdisoltani et al. [14] | **51.38** | - |

Table 2. Classification accuracy (in percent) on the Something-something data set[8] with 174 label categories.

## 5.2. Results

We test the PredNet+ architecture on our best 4 layer model, 5 layer model and 6 layer model from Table 1. The 5 layer model performs better than the 4 layer one, while the 6 layer model, Model 7, even fails to converge. We derive that this is because of the increased depth and parameters in the network. However, it is not clear if this is because of dying-out of gradients in the network or because of the hierarchical error minimization framework of PredNet. Furthermore, we test minor variations of the PredNet+ to further evaluate the model architecture. First, we remove the recurrent memory in the label classification unit by replacing the ConvLSTM with Convolution layers. Next, we extend the label classification loss function such that the model is rewarded for predicting at least the correct verb in the label. For example, if the correct label is *Pretending to put something behind something*, the model is penalized twice as much if it predicts *Showing something to the camera* than if it predicts *Putting something behind something*, which has the same verb as the correct label. Surprisingly the classification results do not change at all ($\pm 0.6\%$) for any of these model variations. This suggests that the features from the top-most representation unit do not have any more information. Table 2 shows our best classification accuracy in comparison to the baseline model scores of Goyal et al. [8] and the current state-of-the-art results by Mahdisoltani et al. on the Something-something data set[14]. Our label classification score suggests that the PredNet+ is a long way from the state-of-the-art architectures. Furthermore, the future frame prediction of the PredNet+ also degrade in comparison to its equivalent vanilla PredNet models: Model 5 ($L_0$ loss) and Model 8 ($L_{all}$ loss). The metrics in Figure 12 and the visualization of predictions clearly point this out. To further analyze this, we experiment with different loss weights for the two tasks. This allows us to control the relative importance of each task for the model during training. We find that the model's future prediction quality degrades when the label classification task is given increased importance, suggesting that the multi-task constraint actually leads to worse future frame predictions.

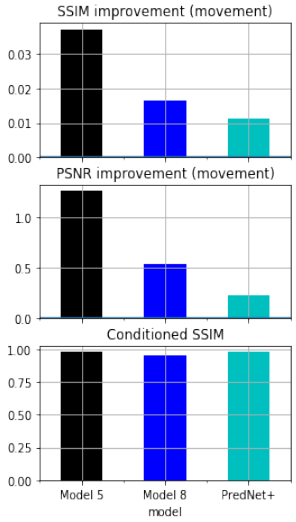


Figure 12. Comparison of the prediction quality of the PredNet+ with equivalent PredNet models.

## 6. Conclusion and future work

In this paper, we have evaluated the PredNet [13] on a challenging action classification data set. Next, we combined the unsupervised video prediction with a supervised video classification task. We have implemented a model that we informally name PredNet+ and trained it in a multi-task fashion. The PredNet+ outputs not only classification results, but it also conditions future predictions on its previously predicted class labels.

In the first phase of our work, we investigate the PredNet and derive the following insights: (1) The PredNet does not completely follow the principles of the predictive coding framework. (2) Its top-down connections are reactive to short-term movements between video frames, i.e. they learn to perform only short-term pixel interpolations, rather than long-term predictions. This has been further confirmed by the extrapolation experiments. (3) The representation units are unable to learn multi-modal distributions and produce blurry predictions as they regress to the mean. (4) The models' learning ability is sensitive to the continuity of motion and the FPS rate of the videos and they perform actual predictions only when the higher layers have full receptive field.

In the second phase, we briefly test the PredNet's ability to learn useful latent features to perform label classification. We use the features from the highest representation layer and find that this is not adequate for the task at hand. We achieve a classification accuracy of 28.2% in comparison to current state-of-the-art of 51.38% [14]. We conclude that further experimenting is still required in this direction.

The above discourse brings forth a lot of scope for future research. A successor to the PredNet can be designed, which does not have the aforementioned limitations and is a more accurate implementation of the predictive coding theory. Firstly, the network should be trainable with $L_{all}$ loss. This can be done by designing error estimators that are local to their layers, i.e. they should assess and penalize the contributions of only their own top-down representation unit. Secondly, the network should be redesigned such that it is encouraged to perform long-term predictions rather than just frame-to-frame interpolation. One way to do this is to have explicit layers, higher in the hierarchy, that make predictions at different temporal scales. Lastly, the estimator units or representation units should learn multi-modal probability distributions, from which predictions can be sampled. Additionally, the PredNet's performance metrics show high variance while the PredNet+ is easily susceptible to over-fitting. These points signal the need for including regularization techniques and model averaging methods like dropout within the architecture. Some of the future work in the PredNet+ would be to connect the 'label classification unit' to representation units of all layers rather than just the top-most layer. In the predictive coding framework, this would be deemed most beneficial. The contribution of label conditioning can be tested by training a model which does not contain this top-down label conditioning. Lastly, on the engineering front, the current implementation of the PredNet takes very long time to train and work can be done towards more efficient usage of GPUs.

## References

[1] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM.

[2] K. De and V. Masilamani. Image sharpness measure for blurred images in frequency domain. *Procedia Engineering*, 64, 12 2013.

[3] J. Deng, W. Dong, R. Socher, L. Li, and and. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[4] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. *CoRR*, abs/1604.06573, 2016.

[5] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 64–72. Curran Associates, Inc., 2016.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[7] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[8] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fründ, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. The "something something" video database

for learning and evaluating visual common sense. *CoRR*, abs/1706.04261, 2017.

[9] K. Han, H. Wen, Y. Zhang, D. Fu, E. Culurciello, and Z. Liu. Deep predictive coding network with local recurrent processing for object recognition. *CoRR*, abs/1805.07526, 2018.

[10] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, June 2014.

[12] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2556–2563, Washington, DC, USA, 2011. IEEE Computer Society.

[13] W. Lotter, G. Kreiman, and D. D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. *CoRR*, abs/1605.08104, 2016.

[14] F. Mahdisoltani, G. Berger, W. Gharbieh, D. J. Fleet, and R. Memisevic. Fine-grained video classification and captioning. *CoRR*, abs/1804.09235, 2018.

[15] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015.

[16] R. Rao and D. H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2:79–87, 02 1999.

[17] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, abs/1506.04214, 2015.

[18] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199, 2014.

[19] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *CoRR*, abs/1502.04681, 2015.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[22] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[23] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. *CoRR*, abs/1704.05831, 2017.

[24] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *CoRR*, abs/1504.08023, 2015.

[25] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. *CoRR*, abs/1608.00859, 2016.

[26] Y. Wang, Z. Gao, M. Long, J. Wang, and P. S. Yu. Predrnn++: Towards A resolution of the deep-in-time dilemma in spatiotemporal predictive learning. *CoRR*, abs/1804.06300, 2018.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 13(4):600–612, 2004.

[28] H. Wen, K. Han, J. Shi, Y. Zhang, E. Culurciello, and Z. Liu. Deep predictive coding network for object recognition. *CoRR*, abs/1802.04762, 2018.

[29] Z. Wu, T. Yao, Y. Fu, and Y. Jiang. Deep learning for video classification and captioning. *CoRR*, abs/1609.06782, 2016.

[30] J. Zhong, A. Cangelosi, X. Zhang, and T. Ogata. Afaprednet: The action modulation within predictive coding. *CoRR*, abs/1804.03826, 2018.

[31] J. Zhong, T. Ogata, and A. Cangelosi. Encoding longer-term contextual multi-modal information in a predictive coding model. *CoRR*, abs/1804.06774, 2018.

# Appendix A: All results

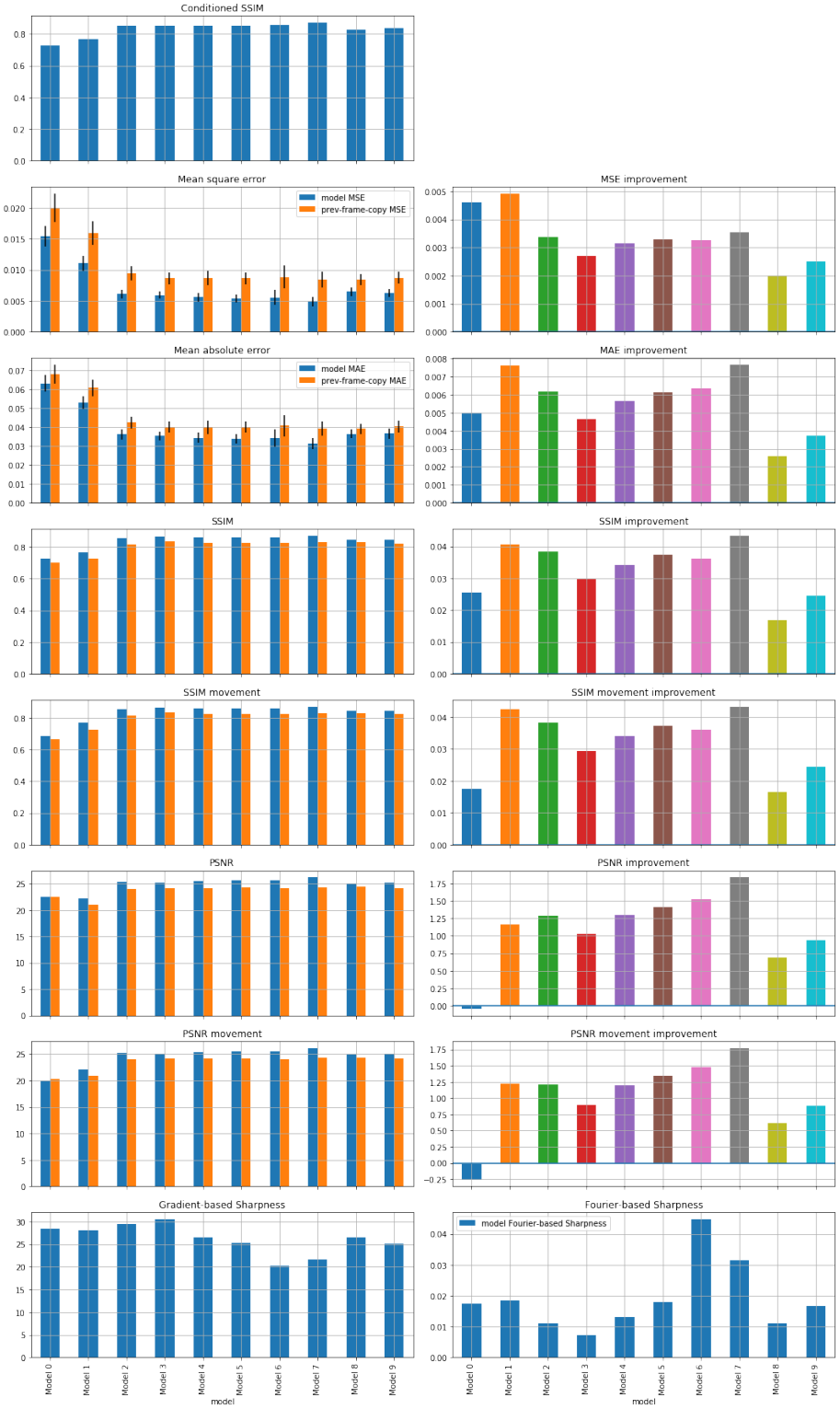


Figure 13. Results of all the metrics from Section 4.1 for all the 10 models listed in Table 1.

## Appendix B: Examples of full visualization

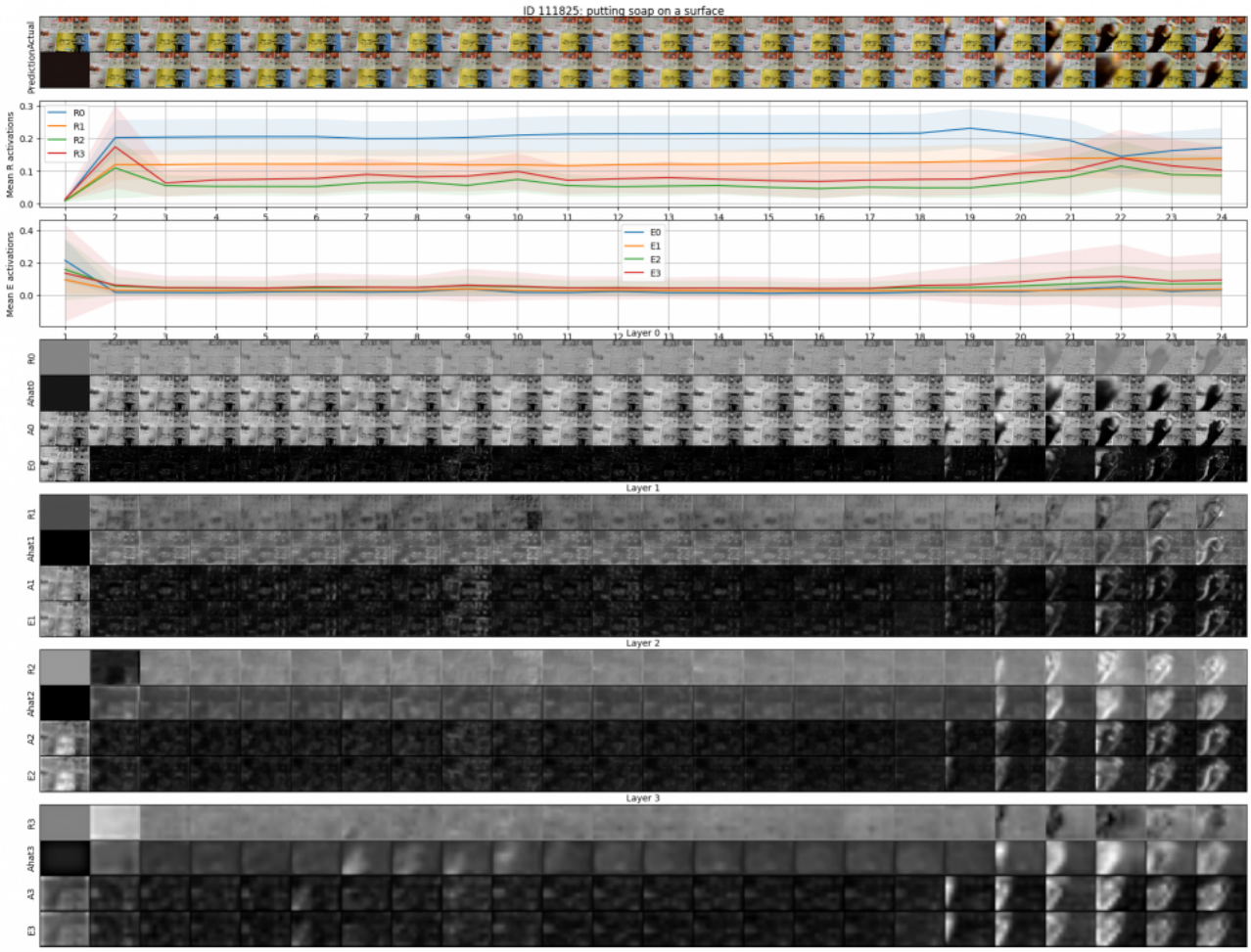


Figure 14. Example of full visualization.

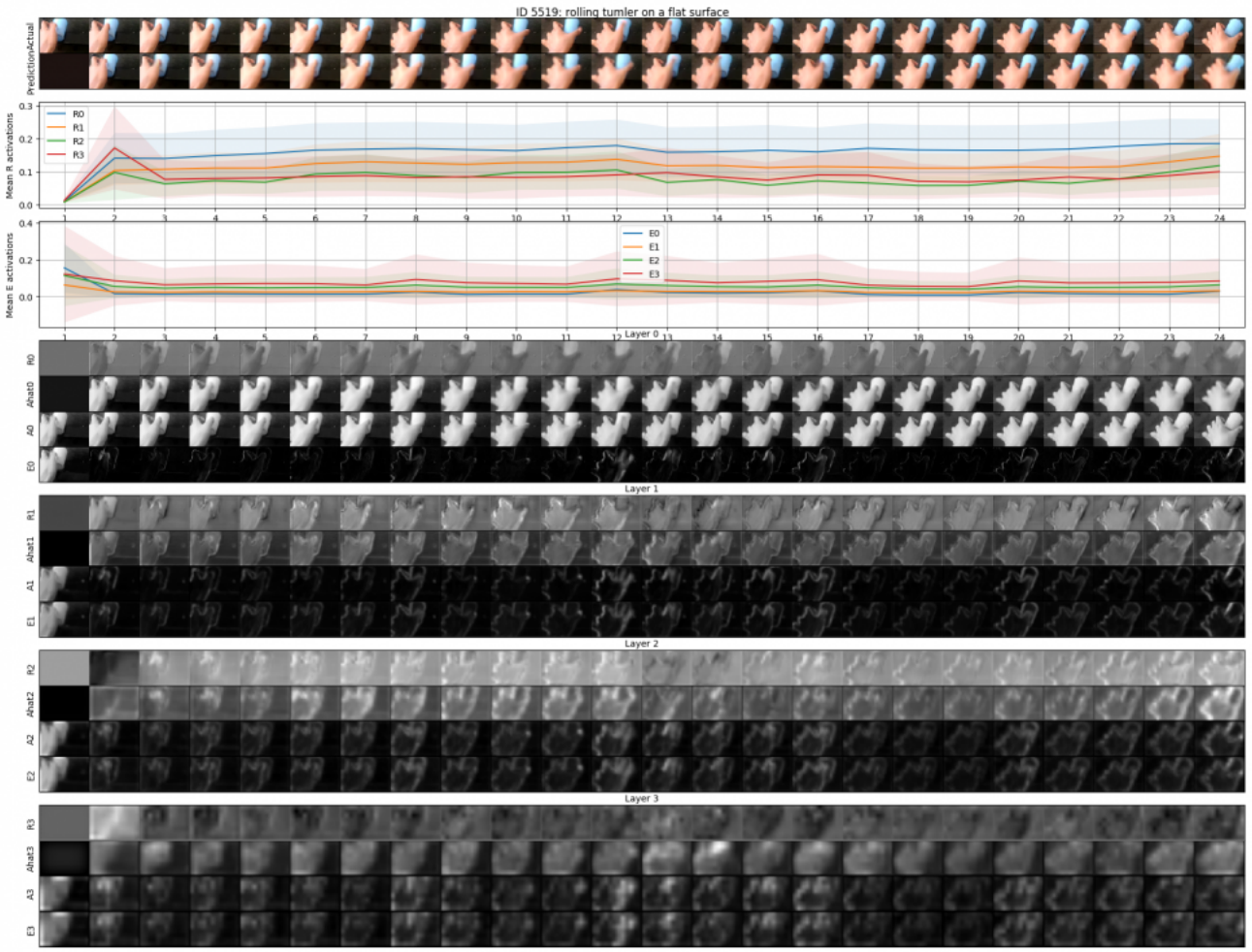


Figure 15. Example of full visualization.

## Appendix B: Example of full visualization on extrapolation.

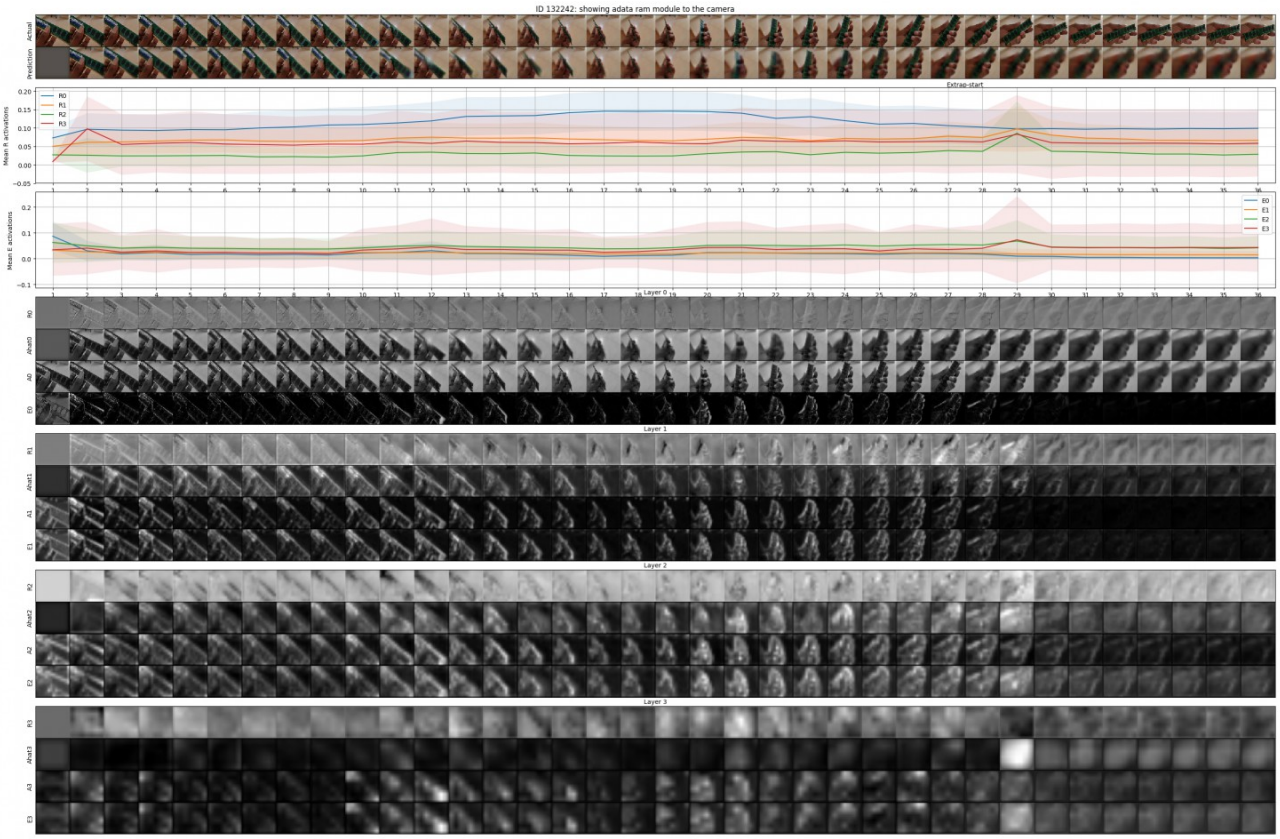


Figure 16. Extrapolation results for Model 7 trained at 3t/4 steps.