Multitask Approach to Video Scene Understanding

VITHURSIKA VINASITHTHAMBY, University of Twente
, The Netherlands

This thesis proposes a multitask approach to enhance video scene understanding by focusing on two distinct, yet complementary, aspects of video content into a finite set of classes. The first task, action recognition, categorises the actions in the video. The second task, object detection, aims to localise any objects in the frame and classify them. We propose a multi-task model that utilises self-attention mechanisms to jointly output action classes, objects, and bounding boxes. The two pre-trained models that encode task-specific information are used as frozen feature encoders to fine-tune the merger model. The approach is evaluated on the EPIC-KITCHENS dataset. This integration is important for maintaining coherent spatial and temporal information crucial for accurate video scene understanding. The multi-task model shows promising results, as it learns well and does not overfit during the training phase. Although the tasks are distinct, leveraging the information leads to a more holistic understanding for each task individually. Additionally, the multi-task model is lightweight, as only a few attention layers are trained. The integration of action recognition and object detection tasks enhances the overall understanding of video scenes, providing a comprehensive and efficient analysis.

Additional Key Words and Phrases: Video Analysis, 3D CNNs, Faster R-CNN, Vision Transformers, Self-Attention Mechanisms, Machine Learning

1 INTRODUCTION

Video scene understanding has significantly improved thanks to advances in deep-learning. Understanding video scenes has a wide range of applications, such as security surveillance, autonomous driving, augmented reality, robot vision, and healthcare. Therefore, it is growing in popularity. Video data is considered to be complex, as it contains both spatial details and temporal information, requiring an approach that integrates both.

2D-CNN models are widely used for image processing purposes; however, they fail to adequately process videos. 3D CNNs are a modification of the classical CNN model, where the kernel convolves over time and space. [18]. Transformers were initially introduced to address complicated NLP problems and have since expanded their use case to other fields, such as computer vision. [19]. Transformers make use of self-attention which allows the model to process the whole input at once. Therefore, it can focus on the most important relations, which enhances the understanding of the input.

R-CNN is a powerful two-step object detection model. The first step entails image segmentation proposals, after which object detection is performed on the extracted regions [13]. As this model lacks the necessary speed, other models have been proposed. Faster R-CNN makes use of Region Proposal Network (RPN), which dynamically proposes object boundaries and thus leads to a more efficient approach [15].

This thesis proposes a holistic understanding of video scenes through a multitask model that:

- Recognises actions within videos
- Recognises corresponding objects related with actions
- Merges the two tasks together to a single scene understanding model

The integration of the two tasks will produce a better understanding of the video. Focusing solely on the action recognition task can lead to an incomplete understanding of video scenes. Particularly with regards to who is performing the action. As it can solely focus on the action, valuable information on who is acting or what object is involved is neglected. On the other hand, only utilising object detection will solely focus on who or what is present in the frame, but fails to capture the nature of actions being performed. Therefore, the combination of the "Action Recognition" task and "Object Detection" task, will lead to a complete understanding of video scenes. The multi-task model will identify actors and objects and interpret their actions, providing a holistic understanding.

Our central question in this work is:

> Can the recognition of actions and action-relevant objects be done concurrently with a multi-task model for holistic video scene understanding?

2 RELATED WORK

2.1 3D CNNs

Traditional approaches for extending 2D-CNNs to videos included a two-stream approach, where temporal and spatial information is processed in two different streams [17]. One stream utilises RGB data and recognises objects and actors, while the other stream processes motion using optical flow fields to capture movement between frames. 3D Convolutional Neural Networks have advanced significantly in video scene understanding. It is important to mention [18], which introduced a method to capture both spatial and temporal features. The advantage of C3D over traditional 2D-CNNs lies in its capability to simultaneously process both visual and motion aspects through uniform convolutional kernels.

A more efficient approach is explored in [9], which introduces the X3D model. The following aspects are considered: temporal duration, frame rate, spatial resolution, network width and depth. The expansion strategy leads to a model that requires significantly fewer resources than earlier models.

2.2 Transformers

Transformers consist of two parts: encoders and decoders. For action recognition, which is a classification task, it is more common to only use the encoder. The Vision Transformer (ViT) adapts the transformer architecture for image processing[7]. ViT segments an

41st Twente Student Conference on IT, July 05, 2024, Enschede, The Netherlands

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

image into patches that are then flattened and embedded. Positional encodings are applied to retain spatial relationships within images. Subsequently, features are represented as queries, with memory embeddings (keys and values) created through linear transformations. The attention weights, which are used to determine the query, are calculated as a dot product between the values and the keys. Subsequently, the transformer encoder processes these weighted values and the classification head outputs the final class probabilities. Video transformers are emerging as a powerful tool in video understanding[1]. As the self-attention mechanism is used, it allows the model to weigh and prioritise certain parts of the video input. On top of spatial encodings, temporal encodings are added to maintain the temporal ordering of the video. Consequently, tokens (patches of the video frames) are processed concurrently using multi-head attention. Multi-head attention differs from single-head attention by processing different features or relationships of the input simultaneously. This allows the model to understand activities and actions, as context from multiple frames can be taken into consideration.

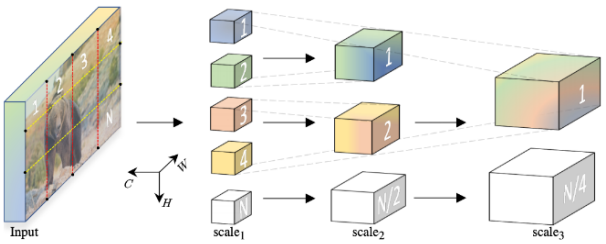

Fig. 1. MViT model [8]

2.3 Two step object detection

Faster R-CNN has two preceding versions: R-CNN and Fast R-CNN. Faster R-CNN introduces the Region Proposal Network (RPN), which significantly accelerates the model. It predicts object bounds and objectness scores through a fully convolutional network[15]. In comparison, for R-CNN and Fast R-CNN, region proposals are generated separately using methods like Selective Search [12, 13]. Faster R-CNN enhances this process by sharing the convolutional network between the RPN and the detection network. Furthermore, Faster R-CNN trains the RPN and detection network separately and fine-tunes them together.

2.4 Existing solutions

The merging of object recognition qualities and action qualities has been discussed in [11]. The Action Transformer model that is introduced makes use of spatiotemporal features to accurately predict the location of individuals and their interaction with their environment. The model puts emphasis on critical regions, such as faces and hands, that are likely to interact with the environment. The model combines the spatio-temporal inflated 3D (I3D) convolutional network with a Region Proposal Network. The model is trained using only RGB frames and shows remarkable results, setting a new benchmark.

The SlowFast network introduced in [10] processes video data at two different speeds. The slow pathway captures the information needed for object detection, whereas the fast pathway captures the information for action recognition. As it processes the data at a higher speed, it will be able to capture motion well.

Temporal Segment Networks (TSN) segments a video into multiple parts and samples snippets from these segments [20]. TSN utilises a two-stream approach where one stream processes the RGB frames and the other stream processes the optical flow.

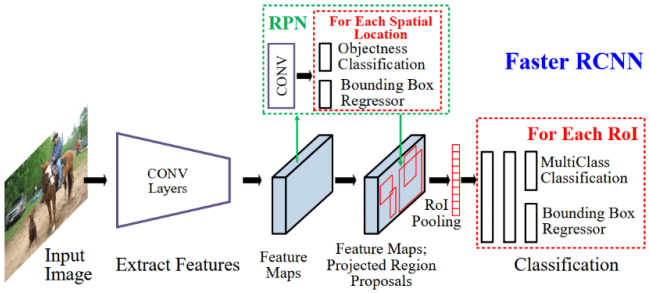

Fig. 2. Faster R-CNN model [15]

3 MULTI-TASK MODEL

Action encoder, this model will provide information as to what action or activity is currently taking place in the video. It considers not only spatial features, but also temporal features. Temporal features show dynamics and motion between frames over time. The MViT model will be used to encode the features [8]. The action encoder is a 3D transformer-based model that uses its self-attention modules to focus on certain temporal features. Furthermore, it is particularly advanced in detecting dependencies over the entire video. The action encoder consists of different layers, such as convolution, pooling and attention layers 1[8]. To encapsulate the features rich in information, the feature map is extracted after the last Multiscale block following the pooling layer.

Object encoder, the faster R-CNN model will be used to encode the features. Object detection demonstrates the objects and actors in the frame, so that we can access the information on who is performing the actions. On top of that, it determines the bounding boxes for the detected objects, such that their location becomes clear. Faster R-CNN is a highly efficient and widely used object detection model that determines both the objects and the bounding boxes[15]. As it is very flexible to its input, it will be easy to apply to the chosen data set. The Faster R-CNN model includes key components, such as convolutional networks, the Region Proposal Network (RPN) and the ROI pooling layer [15]. To capture high-level spatial information, the feature map is extracted from the convolutional layer before the ROI pooling layer.

Multi-task model, this model will integrate the feature vectors from each model and produce classifications. We use a transformer-based model due to its self-attention mechanisms, which are very useful for combining the outputs of two different models. Furthermore, the transformer model is highly flexible and can be applied to the input size and layers needed. The primary goal of the multi-task model is combining the outputs of the object detection and action recognition models into a cohesive output that increases

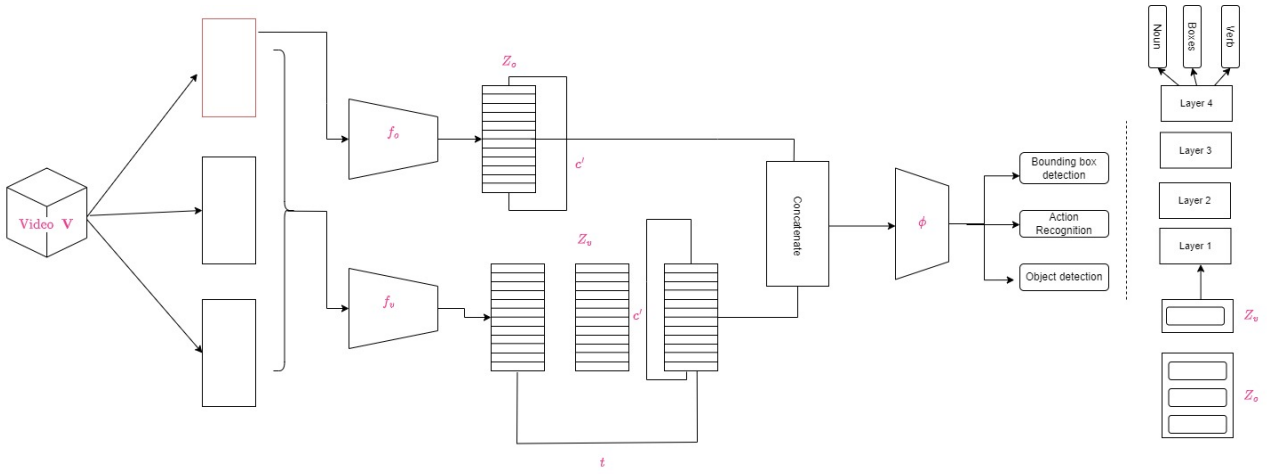

Fig. 3. Multi-task model architecture

Video frames **V** with dimensions Each input video segment **V** has dimensions $\mathbb{R}^{T\times W\times H\times C}$ are used by action encoder f_v to produce spatio-temporal features $\mathbf{Z}_v$ of the dimensions $\mathbb{R}^{t\times w\times h\times c'}$. Similarly, object encoder f_o produces an object related feature map $\mathbf{Z}_o$ of the dimensions $\mathbb{R}^{w\times h\times c'}$. Both action and object encoder features are concatenated and passed to the trainable multi-task model ϕ with four attention layers and three classification layers as shown on the right. The model ϕ outputs three vectors, namely the bounding box coordinates, the probability distribution for the action classes and the probability distribution for the object classes.

the understanding of video scenes. The proposed strategy is to use Faster R-CNN for object detection and to use MViT model for action recognition. Finally, the outputs are combined using a transformer.

3.1 Model Architecture

Each input video segment **V** has dimensions $\mathbb{R}^{T\times W\times H\times C}$. Firstly, this segment goes through the object detection encoder f_0, as shown in Figure 3. The object detection encoder is the Fast R-CNN model which is used as a frozen encoder. We use (video) MViT as our frozen encoder action encoder f_A. Object detection samples one frame at a time, while action detection considers the width W, height H and time T, thus having one more dimension. This produces $\mathbf{Z}_o = f_0(\mathbf{V})$ and $\mathbf{Z}_v = f_A(V)$, where Z_v additionally has c' channels.

The outputs $\mathbf{Z}_o$ and $\mathbf{Z}_v$ are concatenated to a single tensor over the temporal dimension before being treated as input to the multi-task transformer denoted by ϕ. ϕ consists of 4 attention layers and three classification layers. Furthermore, ϕ has 8 heads for each attention layer with a feed-forward dimension of 2048. Let $\hat{\mathbf{Z}} = [\mathbf{Z}_o, \mathbf{Z}_v]$ when concatenated. Then during training of ϕ,

$$\hat{y}_o, \hat{y}_n, \hat{y}_v = \phi(\hat{\mathbf{Z}})$$

where:

- $\hat{y}_o$ represents the bounding box coordinates.
- $\hat{y}_n$ represents the probability distribution for the object class corresponding to a set of nouns.
- $\hat{y}_v$ represents the probability distribution for the action class corresponding to a set of verbs.

3.2 Multi-task objectives

The objective of ϕ is to detect bounding boxes for objects visible in the frames, classify objects visible in the frame and recognise actions in **V**.

To achieve these objectives, two types of loss functions are used:

(1) Cross Entropy Loss (CE):

$$CE_n(\hat{y}_n, y_n) = -\sum_{i=1}^{N} y_{n,i}\log(\hat{y}_{n,i})$$

$$CE_v(\hat{y}_v, y_v) = -\sum_{i=1}^{N} y_{v,i}\log(\hat{y}_{v,i})$$

where $\hat{y}_n$ and $\hat{y}_v$ are the predicted labels for nouns and verbs, respectively and y_n and y_v are the ground truth labels [14].

(2) Mean Squared Error (MSE) Loss:

$$\text{MSE}(\hat{y}_o, y_o) = \frac{1}{N}\sum_{i=1}^{N}(y_{o,i} - \hat{y}_{o,i})^2$$

where $\hat{y}_o$ is the predicted bounding boxes and y_o is the ground truth bounding boxes. The MSE loss measures the average of the squared differences between the predicted value $y_{o,i}$ and the truth value $\hat{y}_{o,i}$ [6].

The final loss obtained to train ϕ is defined as:

$$L = \text{CE}_n(\hat{y}_n, y_n) + \text{CE}_n(\hat{y}_n, y_n) + \text{MSE}(\hat{y}_o, y_o) \quad (1)$$

The final loss combines three losses such that ϕ can learn and optimize the tasks concurrently. This approach ensures that the model classifies $\hat{y}_n$, $\hat{y}_v$, $\hat{y}_o$ accurately.

3.3 Optimisation

To further improve the accuracy of the multitask model, a few small adaptations need to be made:

(1) **Activation layer**: Choosing the right activation layer can impact the accuracy of the model heavily. The activation layer will allow non-linearity through the tanh function in the model, which will help the model learn more complex patterns. It takes the output of the previous layer and modifies the data, such that it can be taken as input for the next layer. For the multitask model, the GELU activation layer was chosen [2].
(2) **Initialise weights**: In the first iteration of the experiment, the weights were not initialised, which caused the model to learn less effectively. To optimise the training process, the classification layers are initialised to values ranging from 0.1 to -0.1. The biases for these layers are initialised to zero.
(3) **Separate loss backward propagation for faster computation**: In the first iteration, the losses for the two classifications and one regression head are combined and then the backward operation is performed at once. However, this approach is computationally very expensive. Therefore, the losses will be computed and the backward operation will be performed on them individually.
(4) **Scheduler**: A scheduler can be utilised to fine tune the learning rate, such that the loss does not stagnate. It will lead to faster convergence and better model performance. The learning rate is slowly decayed at specific intervals with a specific rate. When training a multitask model, it could be the case that one task has stopped improving, but the learning rate is not d. The ReduceLROnPlateau scheduler is designed to reduce the learning rate when a specific task has stopped improving. [3] Therefore, it allows higher learning rates when the model is performing well and small learning rates when it is performing worse.

Bayesian optimisation Hyperparameter tuning is widely used to increase model performance. There is a wide range of hyperparameter optimisers, ranging from grid search to bayesian optimisation. As the multi-task transformer is quite complex, a bayesian optimisation technique is preferred.

Multi-arm bandit is a state-of-the-art optimisation algorithm that finds the most efficient hyperparameters for the model being trained [16]. The multi-arm bandit algorithm can be easily integrated into the training logic that has already been defined. The core components needed to start the tuning of the hyperparameters are:

- Parameters dictionary: This dictionary contains the parameters that will be tuned. The following parameters were chosen with the following space:
  - Attention layers: Adding layers can enhance the model's capabilities to learn complex patterns. However, too many layers can lead to overfitting and therefore negatively impact the model's performance. Furthermore, if too many layers are added, the computational complexity of the model will significantly increase.
  - Optimisation algorithm: Different optimisation algorithms can have different effects on how the model is trained according to the loss function. The choice of optimisation algorithm can influence the convergence speed and the performance of the model.
  - Learning rate: The learning rate is extremely important to reach a high performing model, because it determines the speed the model learns at. If the learning rate is too high, it will cause the model the overshoot and it will not achieve its optimal performance. If the learning rate is too low, it will be computationally expensive to achieve any results.
- Objective specification: This function will evaluate the model's performance with the hyperparameters given for that iteration. The objective function will contain the training of the module. During training the module will take the different parameters and return the summed loss of the action recognition, object detection and bounding boxes.
- Tuner: The tuner will find the best hyperparameters with the use of the minimise and maximise functions. As the objective function will return the loss, it will be task of the tuner to minimise this loss. Therefore, the tuner will be set to the minimise function. After which it will return the best-performing hyperparameters.

4 RESULTS

In section 4.1, we will discuss the evaluation metrics used to measure the model's performance. In section 4.2, we will cover the experimental setup, including variables such as the dataset split and preparation. In section 4.3, we will present the results measured for action recognition, and in section 4.4, we will discuss the results for object detection. Section 4.4 will cover the bayesian optimisation applied to ϕ, along with the results of the model with the optimisation. An ablation study is performed and analysed in section 4.5. Finally, in section 4.6, qualitative analysis is conducted to assess the model's performance further.

4.1 Evaluation metrics

The performance of ϕ was evaluated using multiple metrics to assess the accuracy of $\hat{y_o}$, $\hat{y_v}$ and $\hat{y_n}$.

Validation accuracy indicates the accuracy of the model on unseen data. It specifies the percentage of correct predictions out of the total number of predictions.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{2}$$

The Intersection over Union (IoU) metric was used to assess the accuracy of the bounding boxes relative to the ground truth. The IoU measures the overlap between the predicted bounding box and the ground truth.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{3}$$

Mean Average Precision (mAP) is often utilised to assess the accuracy of object detection models. mAP calculates the average precision for different recall values. AP is the area under the precision-recall curve. Often mAP is used in combination with IoU, where a

prediction is considered correct when the IoU value crosses a certain threshold. The mAP is calculated for the following threshold: 0.25, 0.50, 0.75. For each threshold, the AP values are averaged to get the mAP value.

The Average Precision (AP) for a single class is given by:

$$\text{AP} = \sum_{n} (R_n - R_{n-1}) P_n \tag{4}$$

where P_n is the precision at the n-th threshold and R_n is the recall at the n-th threshold.

The Mean Average Precision (mAP) is calculated as:

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^{C} \text{AP}_c \tag{5}$$

where C is the number of classes and AP_c is the average precision for class c.

Finally, the precision and F1 scores were also used for action recognition and object detection. Precision offers insight into the proportion of true positives among all the made detections. The precision metric will show the number of false positives a model makes. F1 scores combine recall and precision and offers insight into the balance between the false positives and the false negatives.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{6}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

4.2 Experimental setup

The models were run on Jupyter Notebook servers with NVIDIA A10 GPU's. Furthermore, the PyTorch library is used to import f_o and f_v models.

The dataset used for the experiment is the EPIC-Kitchens 55 dataset[5], which consists of a large amount of video data. In total, there are 125 verb classes and there are 325 noun classes. The dataset contains videos of actors cooking recorded with a camera. The dataset contains annotations, not only for the actions performed and the objects present in the frames, but also for the bounding boxes. The EPIC-Kitchens dataset contains both a bigger 100 and a smaller 55 version. Due to resource constraints, the EPIC-Kitchens 55 was chosen. 80% of the dataset with ground truth is loaded as the training dataset, whereas 20% is loaded as validation data.

The frames of the video will be resized to 224, and a center crop will be performed to 224.

Furthermore, for each annotated action, 16 frames were uniformly sampled, as that is the required input for f_v. From the 16 samples that were sampled, 4 frames were subsampled for f_o.

$\hat{Z}$ is normalized by simply taking the mean. Finally, it is converted into a pickle file, which is saved in the dataset. After all the $\hat{Z}$ have been aggregated, it is loaded using a custom dataset and a dataloader.

The Adam Optimiser (Adaptive Moment Estimation) has gained a lot of attraction for its high performance with respect to deep learning models [4]. The Adam optimiser computes adaptive learning rates and includes bias correction.

The model is initialised to the following hyperparameters:

(1) Learning rate: 0.0001
(2) Batch size: 100
(3) Number of Epochs: 20
(4) Optimizer: Adam
(5) Dropout rate: 0.1

4.3 Action Recognition

In Table 1 it can be seen that the highest accuracy for action recognition (denoted with verb) is 41%, which is a slight improvement when compared to the baseline of 40%. Precision and F1 scores are slightly under the baseline. This suggests challenges with recall, resulting in fewer correctly identified positive cases.

The validation loss continuously decreases for action recognition in Figure 4, which shows that the model is still learning until the 20th epoch. The model is not overfitting until then as the loss is decreasing, which shows the ability of the model to generalise to new unseen data.

Metric	Val Acc Verb	Val Precision Verb	Val F1 Verb
Highest Value	0.41	0.34	0.34
Baseline Verb	0.40	0.40	0.38

Table 1. Validation Metrics for Verb-Related Performance

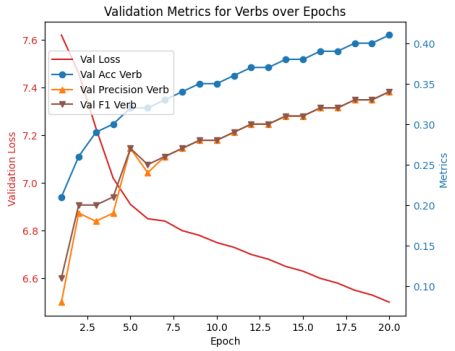

Fig. 4. **Performance action recognition over 20 epochs.** Metrics for the first 20 epochs on the EK-55 val set. The validation loss is indicated with red. The accuracy, precision and f1 scores are denoted with blue, orange, brown respectively.

4.4 Object Detection

The highest validation accuracy for object classification is 16% compared to the baseline of 27% which is a significant decrease as can be observed in Table 2. Furthermore, there is also a significant decrease for precision and the F1-score compared to the baseline.

There is a huge difference between the accuracy, precision, and F1 score of the action recognition tasks and object detection tasks. The difference can be explained by the fact that there is a huge difference in amount of classes for the two tasks. The action recognition tasks only has 125 classes, whereas the object detection class has 325 classes. As object detection task is more complex, it leads to training over more epochs to obtain better performance.

Metric	Val Acc Noun	Val Precision Noun	Val F1 Noun
Highest Value	0.16	0.13	0.14
Baseline Noun	0.27	0.29	0.27

Table 2. Validation Metrics for Noun-Related Performance

The validation loss is plotted in red in Figure 5 and can be seen decreasing consistently from epoch 1 to epoch 20. The graph shows a small plateau around the 5th epoch, which quickly turns into a descending loss again. This indicates that the model is still learning and improving on the dataset.

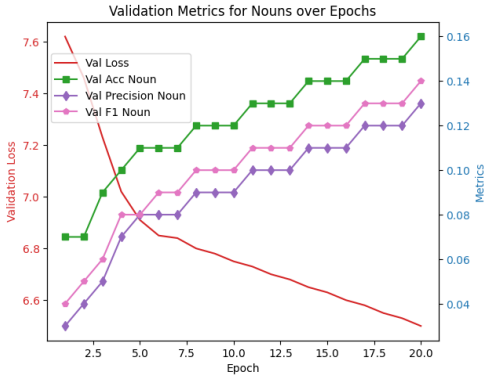

Fig. 5. **Performance object detection over 20 epochs.** Performance metrics for the first 20 epochs. Red represents the validation loss. Green, purple, and pink, respectively, indicate the accuracy, precision, and F1 scores.

The highest value for the mAP value is 0.06, measured at the 19th epoch, which is an improvement on the baseline in Table **??**. The mAP has been showcased for different thresholds over 20 epochs 6. The blue line showcases the mAP value for the 0.25 threshold. It can be seen that the model is consistently learning and the predictions for the bounding boxes are getting better. However, the red line, showcasing the 0.75 threshold, does not show any significant improvements.

The mAP for the bounding boxes is quite low in Table 3. However, it can be seen that the graph has an increasing trend until approximately the 20th epoch. The low precision at the beginning indicates the model struggling to localise the bounding box coordinates and therefore resulting in a low IoU. As the multi-task transformer is quite simple, it is hard for the model to learn more complex structures. Utilising weighted losses or a combination of losses could lead to a better performance.

Metric	mAP@0.25	mAP@0.5	mAP@0.75
Baseline	0.00	0.00	0.00
Multi-task	0.06	0.03	0.00

Table 3. AP and mAP values for object detection performance.

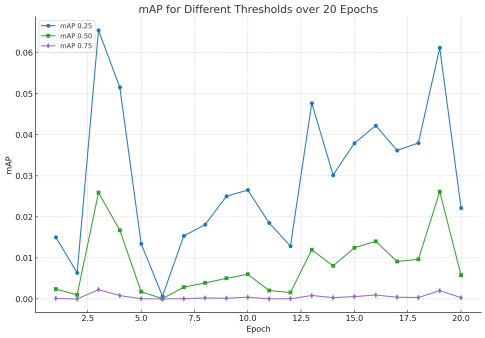

Fig. 6. **mAP over IOUs thresholds for object detection.** Metrics for the first 20 epochs on the EK-55 val set. Thresholds $\{0.25, 0.5, 0.75\}$ are used for the IOU between predicted and ground truth boxes.

4.5 Results Bayesian Optimisation

The results of the bayesian optimisation algorithms are showcased in Table 7 and Figure 8. The bayesian optimisation algorithm Multi-arm bandit was utilised to find the right number of attention layers, the right learning rate and optimisation algorithm. It can be seen in Figure 7 that there is a slight decrease in the loss when the amount of attention layers increases. Furthermore, as the learning rate increases in Figure 8, the loss also slightly increases.

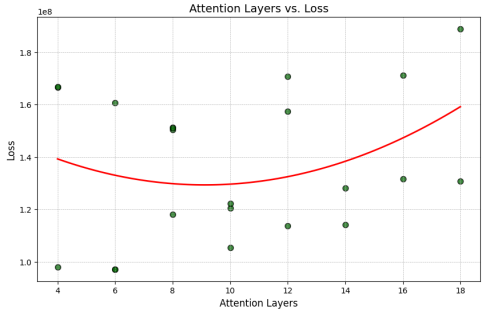

Fig. 7. **Attention Layers vs Average Loss** Performance measured by the loss as the attention layers increase as the result. The trend line is represented by the blue line and can be seen increasing as the number of layers increases.

In Figure 7 the lowest losses can be measured around layer 4 and layer 6. However, ϕ already consist of 4 layers meaning that it is already optimised.

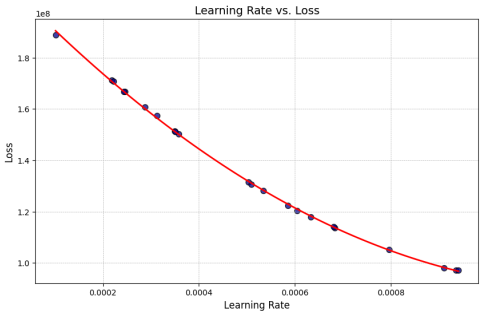

Fig. 8. **Learning Rate vs Average Loss**
The trend line is represented by the blue line, and it shows that increasing the learning rate increases the loss per epoch.

Figure 8 shows that 0.009 causes ϕ to have the smallest loss. Therefore, after adjusting the hyperparameter learning rate to 0.009, the following results were obtained.

The results are summarised in Table 4. It is surprising that ϕ performs worse when trained with the learning rate 0.009. Furthermore, the validation loss reported is higher than reported earlier. This can be explained by the fact that the steps are simply too big and therefore it misses the minima.

4.6 Ablation

Table 5 summarises the results of the ablation studies evaluated over a multitude of metrics. Each ablation removes or modifies an element of the model, such that the impact of that particular element can be tested.

The first entry in the table showcases the model as it is working.

The second entry in the table indicates the reducing of the transformer layers from 4 to 2. The reduction of the attention layers does not show any significant changes in the performance of the model. Therefore, the model seems to be extremely lightweight, needing only two attention layers to achieve similar performance.

Moreover, when removing the dropout and activation functions, we can observe a significant decrease in all metrics. When no dropout is applied, the changes of the model overfitting on the training data will increase. Furthermore, without an activation layer, it will be harder for the model to learn non linear representations.

Furthermore, the model was run for 50 epochs. When the model is trained for a longer period, it can be seen that the performance does not get better. This indicates that the model is not learning and therefore it is overfitting on the training data.

The use of dropout in the multi-task transformer should avoid the case of overfitting, as it randomly sets input neurons to zero and therefore makes sure that the model learns robust features. Even with the use of the dropout function, overfitting can be observed. This could be due to several reasons, such as a too low dropout rate or a necessity for other regularisation techniques.

Finally, removing the scheduler from the multi-task model does not seem to have any significant impacts on the performance of the model as well. However, all the metrics decrease slighty, which indicates the learning rate adjustments have effect. It could be that a different learning rate scheduler must be choosen or the parameters need to be improved for better performance.

4.7 Qualitative Analysis

In order to analyse on what frames the model performs well, four frames have been selected, shown in Figure 9, that were correctly classified by the model and four frames were chosen that were incorrectly classified.

For Action Recognition, the frames that were identified (9a, 9b) correctly are well-lit and the action is happening in the middle of the frame. The poor classification of frames (9g, 9h), can be attributed to the poor visibility of the action due to bad lighting conditions. Furthermore, the actions predicted well are often recurring actions in the dataset. Therefore, the model will be able to detect it better. Finally, some actions are partially disturbed due to either pre-existing factors or the center-crop being performed.

For Object detection, ϕ performs significantly better when it is detecting an object that is in a well-lit environment 9d, ϕ performs well. However, 9f shows a poorly angled frame shot and therefore ϕ struggles to classify the objects. 9e is a poorly lit frame and therefore the objects are detected correctly.

5 FUTURE WORK

ϕ shows difficulties adapting to frames that have poor lighting, and therefore data preprocessing methods will be helpful. For example, data augmentation techniques can be applied to improve the robustness and the ability to generalise. Furthermore, augmenting the dataset to address underrepresented classes will enhance the model's ability to generalise and improve performance.

6 CONCLUSION

This thesis explored the possibility of combining action recognition with object detection using a multitask model. This study used the EPIC-KITCHENS 55 dataset, and Faster R-CNN and MViT for preprocessing the data. Experimental data shows improvement over 20 epochs, as the validation loss decreases consistently.

However, there are significant differences between the two tasks. The action recognition task with 125 classes is more difficult than the object detection task with 325 classes. The reason for this can be accredited to the added complexity that comes with more classes.

The mAP values over the three thresholds remain low, which shows the inability of the multi-task model to accurately localise the object. Despite these challenges, the mAP showcased that there is an increasing trend. It indicates the model is learning and localising objects better.

The added complexity of object detection in the form of extra classes and having to localise and classify the objects is a challenge. Balancing the loss between the action recognition task and the object detection task could be useful to address the challenge with object detection. For example, weighted losses where the object detection loss has a higher weight can help the model learn the features for this task better.

The research question defined above is the following.

	Validation Loss	Validation Accuracy	Validation Precision	Validation F1 Score
Verb	7.46	0.22	0.06	0.10
Noun	7.46	0.05	0.01	0.01

Table 4. Highest Values for Verb and Noun Metrics

> Can the recognition of actions and action-relevant objects be done concurrently with a multi-task model for holistic video scene understanding?

From the experimental results, it can be concluded that it is possible to combine action recognition and object detection using a multi-task model. The concurrent recognition of actions and action-relevant objects was achieved through Faster R-CNN and MViT. The multi-task model is defined to be a transformer model utilising self-attention mechanisms. The model demonstrated the ability to learn and improve over time.

However, the noun-related performance metrics (accuracy, precision, and F1 score) showed significant declines compared to their baselines, highlighting difficulties in noun classification. Additionally, the action recognition metrics showed only slight improvements in accuracy and declines in precision and F1 scores. These difficulties occur because the model is optimising three tasks at once. Although there is potential, further refinement is required to calibrate the tasks into one coherent objective. However, the object detection metrics showed significant decline compared to the baselines, and therefore indicating the poor performance. Additionally, the action recognition metrics only showed slight improvement compared to the baseline. Thus, small adjustments and refinements are necessary for the multi-task model to be accurate. Augmenting the dataset to address underrepresented classes and poor-quality frames can help increase the model's performance.

7 REFERENCES

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. 2021. ViViT: A Video Vision Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 6836–6846.

[2] Baeldung. 2024. GELU Activation Function. https://www.baeldung.com/cs/gelu-activation-function Accessed: 2024-06-14.

[3] Aleksey Bilogur. 2021. LR Schedulers, Adaptive Optimizers. https://residentmario.github.io/pytorch-training-performance-guide/lr-sched-and-optim.html Accessed: 2024-06-14.

[4] Jason Brownlee. 2021. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/ Accessed: 2024-06-14.

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2020. The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020).

[6] DataCamp. 2023. Loss Functions in Machine Learning Explained. (2023). https://www.datacamp.com/tutorial/loss-function-in-machine-learning Accessed: 2024-06-15.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[8] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. 2021. Multiscale Vision Transformers. *arXiv preprint arXiv:2104.11227* (2021).

[9] Christoph Feichtenhofer. 2020. X3D: Expanding Architectures for Efficient Video Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. SlowFast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6202–6211.

[11] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. 2019. Video Action Transformer Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 244–253. https://openaccess.thecvf.com/content_CVPR_2019/papers/Girdhar_Video_Action_Transformer_Network_CVPR_2019_paper.pdf

[12] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*. Microsoft Research.

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE.

[14] Kurtis Pykes. 2024. Cross-Entropy Loss Function in Machine Learning: Enhancing Model Accuracy. https://www.datacamp.com/tutorial/the-cross-entropy-loss-function-in-machine-learning. Accessed: 2024-06-14.

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in neural information processing systems*. Microsoft Research.

[16] Sandeep Singh Sandha, Mohit Aggarwal, Igor Fedorov, and Mani Srivastava. 2020. Mango: A Python Library for Parallel Hyperparameter Tuning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3987–3991.

[17] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*. 568–576.

[18] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *Proceedings of the IEEE international conference on computer vision*.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*. Google Brain, Google Research, University of Toronto.

[20] Chao Wu, Yuecong Xu, Xiaojun Xu, Wei Wang, and Dacheng Tao. 2019. Baidu-UTS Submission to the EPIC-Kitchens Action Recognition Challenge 2019. arXiv:1906.09383 [cs.CV]

Ablation	Verbs				Nouns				mAP@0.25
	Accuracy	Precision	F1 Score	Loss	Accuracy	Precision	F1 Score	Loss	
Multi-task Model	0.41	0.34	0.34	6.50	0.16	0.13	0.14	6.50	0.06
Reduce to two Attention Layers	0.36	0.37	0.34	6.36	0.16	0.14	0.14	6.36	0.03
Remove Dropout + Activation	0.22	0.07	0.10	7.50	0.05	0.00	0.01	7.50	0.0007
Increase to 50 epochs	0.36	0.38	0.34	6.60	0.15	0.16	0.15	6.60	0.03
Remove Scheduler	0.36	0.33	0.31	6.48	0.14	0.12	0.12	6.48	0.04

Table 5. Action Recognition Metrics for Different Ablation Studies

A ADDITIONAL DETAILS

Due to cropping of the images, it can be the case that no bounding boxes are available for that set of frames. If this is the case, then a tensor of zeros will be outputted. The bounding boxes are only considered when they are fully visible in the cropped frame.

B STATEMENT

During the preparation of this work, the author(s) used Grammarly in order to check spelling and grammar, LaTeX for text composition, and Zotero for citation management. After using these tools/services, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the work.

C ABLATION

D QUALITATIVE ANALYSIS

(a) Well performing frame for object detection.

(b) Well performing frame for object detection.

(c) Well performing frame for action recognition.

(d) Well performing frame for action recognition.

(e) One of the frames that performs less when used for object detection.

(f) Frames with less performance with regards to object detection.

(g) Frame with less performance for action recognition.

(h) Less performing frame for action recognition.

Fig. 9. Performance of frames in object detection and action recognition