CLASSIFY HAPPY AND SAD IMAGES WITH VISI AND MASK ATTENTION MECHANSIM

NGUYEN DAC HOANG PHU, NGUYEN ANH MINH, NGO DUY THINH,
PHAN THANH DAT, VO DOAN THINH
VIETTEL DIGITAL TALENT PROGRAM 2023

ABSTRACT

Image classification is an important task in computer vision, with various applications in fields such as healthcare and security. In this paper, we propose a Vision Transformer model with masked attention for image classification of two classes: tired and awake. At the same time, we evaluate the performance of our model and compare it with other popular image classification models. The results show that our Vision Transformer model has the same results as the ResNet50 model and better than the other models, achieving about 97% accuracy on the test set. Our findings demonstrate the effectiveness of using a Vision Transformer model with masked attention for fatigue image classification.

KEYWORDS

Vision Transformer, Mask Auto Encoder, Image Classification

1 INTRODUCTION

Image classification is one of the important problems in the field of Artificial Intelligence. It is widely applied in various fields such as healthcare, security, image processing, computer vision, etc. With the development of technology and data, machine learning models are increasingly being improved to solve more complex problems. Currently, monitoring and assessing the fatigue and alertness of humans is very important. It is applied in solving essential human issues such as diagnosis and treatment of related pathologies, managing and optimizing work productivity to prevent unwanted situations, detecting and warning the fatigue status of traffic participants to increase safety and minimize the risk of traffic accidents.

This research focuses on applying the Transformer model - one of the newest machine learning models today, to solve the image classification problem with a dataset consisting of 2 labels: tired and awake. The aim of the study is to explore the ability of the Transformer model in image classification, and to apply some Attention mechanisms, and finally to compare the results with other classification models.

Use-case. The team's goal is to apply the Vision Transformer model to the image classification problem and compare it with other models.

- Develop ViT model to classify face images based on 2 labels: tired and awake.
- Install some well-known models in the image classification problem for comparison purposes.
- Visualize the results graphically and analyze the methods used in the models.

Supervised by Dr Nguyen Van Nam.

Project2 in VDT, gen 3, (2023)

Related Work.

- Analyze problem requirements and make appropriate data collection choices.
- Classify and find methods to improve the quality of the dataset.
- Experiment with different models to get an overview of the ViT model.
- Conclude and provide meaningful insights into the models.

2 DATA PREPROCESSING

2.1 Data collection

The data set we selected for this study are two classes happy and sad in the AffectNet dataset. It is an image dataset created for research on emotion recognition. This dataset includes more than 1 million face images labeled with emotions, including 8 main emotions: neutral, happy, angry, sad, fear, surprise, disgust, contempt. The AffectNet dataset was created by aggregating from a variety of sources, including earlier datasets such as FER2013 and Emotionet, as well as photographs collected from the Internet. The photos are curated and labeled by experts in the field of emotion recognition.

This data set is customized and posted on Kaggle, we use 2 classes happy and sad with the total number of images in the happy class is about 7000 images and in the sad class is about 5000 images, the amount of data between the 2 classes there is a relative difference but still can not cause the problem of data imbalance. All input images have the same size of 96x96. Although the problem is to classify images based on 2 types, tired and awake, we could not find the exact data set compared to the request. Therefore, we decided to choose two alternative equivalence classes, happy and sad, which have the highest correlation and are most feasible to conduct research on the image classification problem.

2.2 Data Augmentation

Data augmentation is an important technique in machine learning and deep learning, especially in image classification. Data augmentation allows to increase the amount of training data by creating new versions of the training data by altering, rotating, cropping or transforming the original images. In this study, we use some popular data augmentation methods as follows:

- Normalization: A Normalization layer should always either be adapted over a dataset or passed mean and variance.
- Rotation: Rotate the image to a certain angle to create new versions of the image.
- Flip: Flip images vertically or horizontally to create new versions.
- Scaling: Resize the image to create new versions with different sizes.

- Crop: Crop part of the image to create new versions with different resolutions.

These data augmentation methods can be applied independently or in combination to create new versions of the training data. The use of these data augmentation methods enhances the diversity and quantity of training data, thereby improving the accuracy of the image classification model.

3 IMPLEMENTATION

3.1 Introduction to Vision Transformer

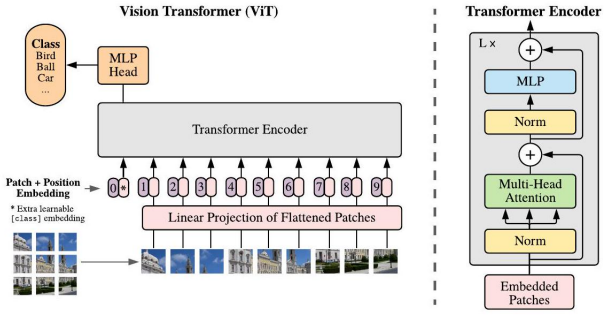

Figure 1: Model overview

An overview of the model is depicted in Figure 1. The standard Transformer receives as input a 1D sequence of token embeddings. To handle 2D images, we reshape the image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ into a sequence of flattened 2D patches $\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where (H, W) is the resolution of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches, which also serves as the effective input sequence length for the Transformer. The Transformer uses constant latent vector size D through all of its layers, so we flatten the patches and map to D dimensions with a trainable linear projection (Eq. 1). We refer to the output of this projection as the patch embeddings.

Similar to BERT's [class] token, we prepend a learnable embedding to the sequence of embedded patches $\left(\mathbf{z}_0^0 = \mathbf{x}_{\text{class}}\right)$, whose state at the output of the Transformer encoder $\left(\mathbf{z}_L^0\right)$ serves as the image representation y (Eq. 4). Both during pre-training and fine-tuning, a classification head is attached to $\mathbf{z}_L^0$. The classification head is implemented by a MLP with one hidden layer at pre-training time and by a single linear layer at fine-tuning time.

Position embeddings are added to the patch embeddings to retain positional information. We use standard learnable 1D position embeddings, since we have not observed significant performance gains from using more advanced 2D-aware position embeddings (Appendix D.4). The resulting sequence of embedding vectors serves as input to the encoder.

The Transformer encoder (Vaswani et al., 2017) consists of alternating layers of multiheaded selfattention (MSA, see Appendix A) and MLP blocks (Eq. 2, 3). Layernorm (LN) is applied before every block, and residual connections after every block (Wang et al., 2019; Baevski & Auli, 2019).

3.2 Cross-Attention Vision Transformer

CrossViT is a type of vision transformer that uses a dual-branch architecture to extract multi-scale feature representations for image classification. The architecture combines image patches (i.e. tokens in a transformer) of different sizes to produce stronger visual features for image classification. It processes small and large patch tokens with two separate branches of different computational complexities and these tokens are fused together multiple times to complement each other.

Fusion is achieved by an efficient cross-attention module, in which each transformer branch creates a non-patch token as an agent to exchange information with the other branch by attention. This allows for linear-time generation of the attention map in fusion instead of quadratic time otherwise.

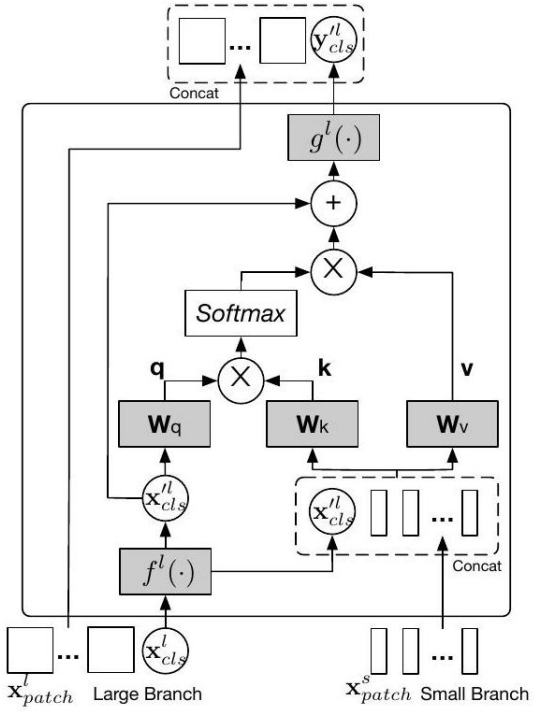

Figure 2: Cross-attention module

In this study, we use two different image patching sets to solve the problem of image classification when the face is tilted or covered.

- Block I: Corresponds to patching the image into several parts to learn all the features in the image.
- Block II: Corresponds to patching the image into few parts (4-6 parts) to learn the areas corresponding to the eyes, nose and mouth.

Then, proceed for the above 2 Patching blocks through Transformer Encoder and finally Cross Attention together to get the Query corresponding to Block II (Learning features of eyes, nose, mouth) and get the Key, Value corresponding to Block I to learn the whole image.

3.3 Masked Auto Encoding in Vision Transformer

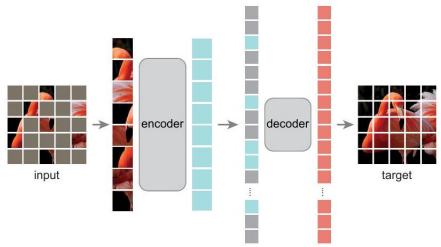

Figure 3: MAE architecture

Masked Auto Encoders (MAE) are scalable self-supervised learners for computer vision. Our MAE approach is simple: we mask random patches of the input image and reconstruct the missing pixels. It is based on two core designs. First, we develop an asymmetric encoder-decoder architecture, with an encoder that operates only on the visible subset of patches (without mask tokens), along with a lightweight decoder that reconstructs the original image from the latent representation and mask tokens. Second, we find that masking a high proportion of the input image, e.g., 75%, yields a non-trivial and meaningful self-supervisory task. Coupling these two designs enables us to train large models efficiently and effectively: we accelerate training (by 3x or more) and improve accuracy.

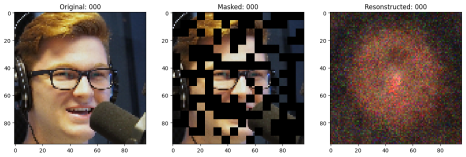

Figure 4: Example results on validation dataset

In figure 4, which is an example showing the functionality of MAE, We show the ground-truth (left), the masked image (middle), MAE reconstruction (right). The masking ratio is 50%.

3.4 Proposed architecture

3.4.1 Pipeline. With the goal of the given problem and the desire to be different from other groups, our main idea in this problem is summarized in Figure 5. Initially, with the input face image, we I pass it through a CNN Encoding block. This CNN block has been trained to be able to Facial Landing Mark on the face of the input image. Then pass it all through the Cross Vision Transformer to generate the Values, Keys and Query. Here, we use 2 different Patching sets. Query will correspond to the large Patching set corresponding to 4 parts of the face, eyes, nose and mouth, and Key and Value will correspond to the small Patching set corresponding to the entire image. Next, we put in MultiHead Attention and then move to the Mask Auto Encoder section. This section has the effect of increasing recognition for images with partially obscured faces. Finally, the output Binary Classification to classify 2 classes as tired and awake.

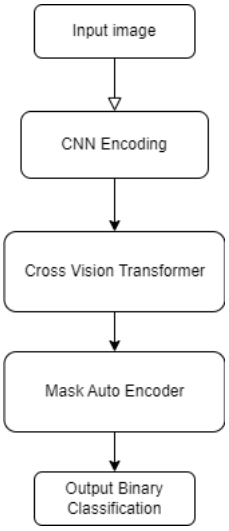

Figure 5: Our Pipeline

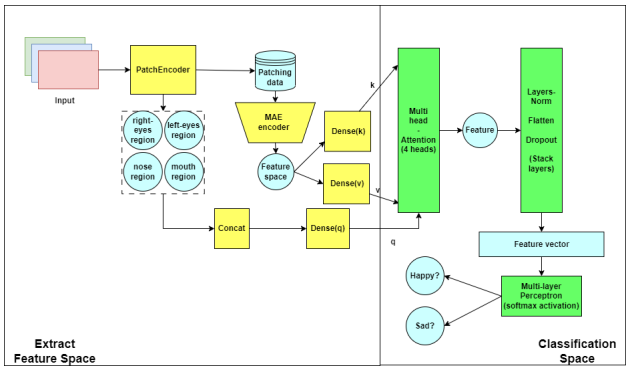

Figure 6: Our Architecture

3.4.2 Architecture. Figure 6 describes our model baseline in more detail. In the Extract Feature section, first, the input image with RGB color system will be put into the PatchEncoder block. Here, it is divided into 2 main tasks:

- Use Facial Landmark Detection to find 4 matrices corresponding to the 4 eye, nose and mouth components. Then feed 4 matrices through dense layer to get same output matrix size. Next, we will concatenate these 4 matrices into a Query (Dense(q)).
- The entire image will be put into the Patching set in the Cross Vision Transformer, then continue through the Mask Auto Encoder (MAE) set to create a Feature space. Finally, from the Feature space will pass 2 different dense layers, one is dense for Key and one is dense for Value.

After obtaining 3 Key, Value and Query matrices. Combine these 3 sets and then move to the Classification section. Here, 3 matrices go through MultiHead Attention (4 heads) to get the Features and then continue to go to stack layers. Then, we get the Feature vector and pass it through the Multi-layer Perceptron (MLP) network with the

softmax activation function. Finally, the output in Binary format determines whether the input image is happy or sad, corresponding to awake and tired according to the requirements of the problem.

4 RESULTS & CONCLUSION

4.1 Summary

Use a CNN layer to extract image features such as eyes, nose, mouth and then pass through Dense Layer Positional Embedding. The goal is to want the model to learn location features during training, thereby increasing prediction performance.

Using MAE to perform the reconstruction image task on the prepared data set in order to help the encoder deduce the hidden space better when encountering cases related to image blur, loss of part of information.

After performing the reconstruction task, get the MAE encoder and use it for task classification, after producing the feature map, we will test the addition of information by adding a query on each specified area and comparing the results.

With the feature map, we will flatten and pass through the Dense layer to get the right information for the problem output.

The model has not achieved high results on f1-score and accuracy compared to other models due to the following reasons:

- There is not enough training data for the reconstruction task, at this time the model has not learned the best representation for the hidden space, in addition, the team has tried to increase the data as well as increase the batch size but due to the limitation of RAM and GPU of colab so this test fails, the team stays the same and accepts to train on the existing data.
- Mask 0.75 can cause loss of information when inferring, the test group is reduced to 0.5 and experiment as shown in the box below.
- Treating this task as a transfer learning can cause the model to have more self-learning feature information like ViT that must depend on MAE learning (which can lead to error propagation).
- The model has higher validation with training because it has a pre-visualization of data information, thereby bringing similar images to lie close together in the hidden space, so that inferring labels on the set better unlearned data.

However, the model also improved as the team increased the dataset (adding the affect young-HQ set), initially only getting an f1-score of around 71%, which shows if a good enough reconstruction of the model can help. for the classifier to achieve high results and increase the inference of the model.

Both ViT and MAE are packed with weights and can be recalled to use for inference without retraining.

According to the results from Figure 8, the model required by the problem has been completed, some comments on this model:

- After querying through the multihead-attention layers, the model flattens and passes through more MLP layers to finally return the label output.
- The use of a fixed query key affects the quality of the model in deducing tilted and rotated models (because at this time the fixed query key has changed). In the data set for inference, the model learned correctly on image sets with face regions according to the query and worse than on regions not according to the query. Suggested improvement is the use of models to recognize parts of the face and generate dynamic queries from the model, thus making the model more complex but will help align the correct positions on the face. face).

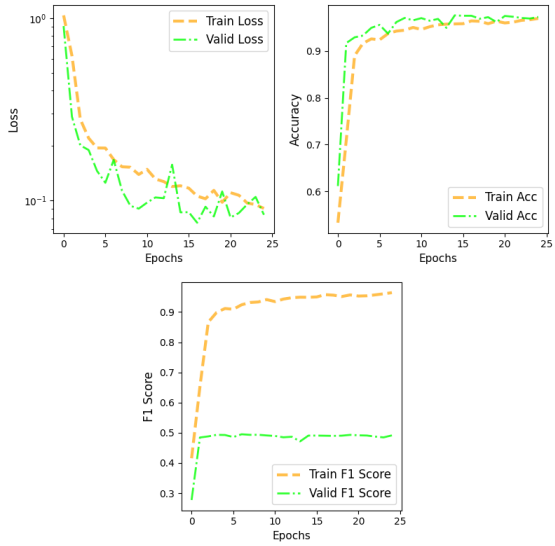

Figure 7: ViT results. Loss (left), Accuracy (right), F1 score (bottom)

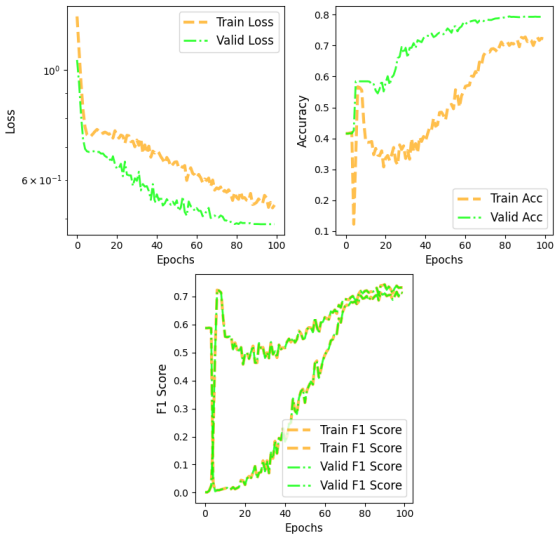

Figure 8: MAEViT (use face query) results. Loss (left), Accuracy (right), F1 score (bottom)

- The model is still the best of MAE in generating good enough hidden space, but the encoder needs more training to make the inference better.

Comparison table:

Model/Res	Accuracy Testset	F1 score Testset
VGG19	58.6	60.2
Res-Net50	98.4	98.1
ViT (Pretrain-model)	97.15	97.06
ViT (Cross-Attension)	97.00	97.10
Proposed Model	97.21	97.15
VIT + ImageGenerator	59.60	40.20
ViT (posEncoding)	98.06	98.00
MAE (0.75% mask / no face query)	85.88	85.29
MAE (0.75% mask / use face query)	77.09	76.96
MAE (0.5% mask)	84.87	84.21

During the experiment, the ImageGenerator tactic was not effective for the problem, after using methods such as rotate, flip, crop, ... to increase the number of training images, the model learns slower and has a better index. worse price than using the DataAugmentation class.

4.2 Inference on new images

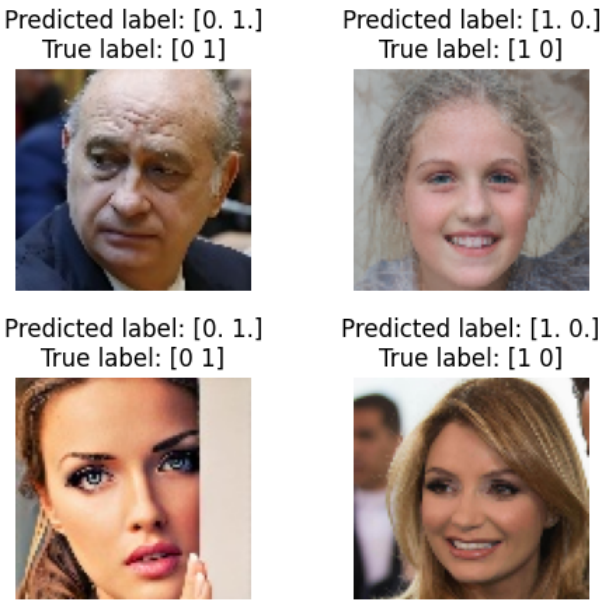

Figure 9: Inference images from test dataset

Conclusion: The model has been completed and met the requirements of the test with a minimum F1-Score of 70%. In the data set for inference, the model learned correctly on image sets with face regions according to the query and worse than on regions not according to the query. However, the model is still the best of MAE in generating good enough hidden space, but the encoder needs more training to make the inference better.

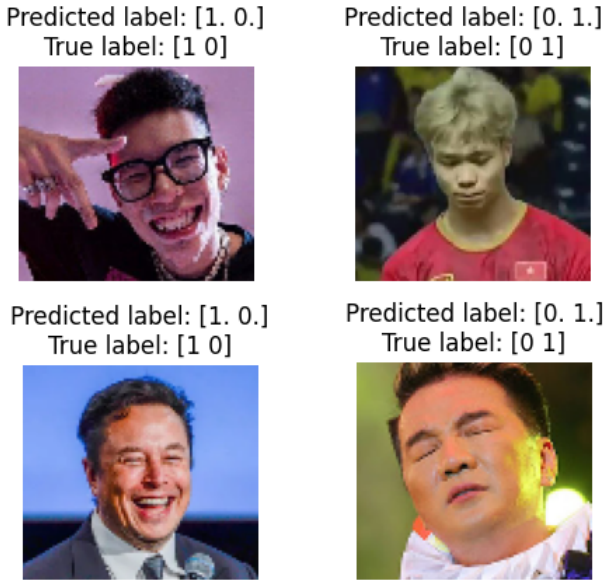

Figure 10: Inference images from new dataset

5 REFERENCES

[1]: **Attention Is All You Need**
https://arxiv.org/abs/1706.03762
[2]: **Formal Algorithms for Transformers**
https://arxiv.org/abs/2207.09238
[3]: **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale**
https://arxiv.org/abs/2010.11929v2
[4]: **Landmark Guidance Independent Spatio-channel Attention and Complementary Context Information based Facial Expression Recognition**
https://arxiv.org/abs/2007.10298
[5]: **CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification**
https://arxiv.org/abs/2103.14899
[6]: **Masked Autoencoders Are Scalable Vision Learners**
https://arxiv.org/abs/2111.06377v2
[7]: **Semi-MAE: Masked Autoencoders for Semi-supervised Vision Transformers**
https://arxiv.org/abs/2301.01431