Deep Learning-based Facial Expression Recognition using CNN and SVM: Insights and Implications

Yiyuan Cui
UCSD
A15438228
Y2cui@ucsd.edu

Naiwen Shi
UCSD
A15554690
nashi@ucsd.edu

Weihua Zhao
UCSD
A14684029
wez205@ucsd.edu

Abstract

Facial expression recognition has emerged as a promising approach for monitoring emotional health and promoting overall well-being. In this paper, we compared the performance of different classifiers, including convolutional neural network (CNN) alone, support vector machine (SVM) alone, and a stacked CNN-SVM model, for the task of classifying facial expressions. A dataset of facial pictures was used to train and evaluate the classifiers, with the stacked CNN-SVM model achieving the highest accuracy of 73%. The paper discusses the implications of these findings and suggests potential future work to improve the effectiveness of facial expression recognition for monitoring emotional health.

1. Introduction

Facial expression recognition has been shown to be a promising approach for monitoring emotional health and promoting overall well-being. Poor emotional health can have a significant impact on physical and mental health, and the ability to accurately detect negative emotions such as sadness and anger can help prevent long-term negative consequences. Recent advances in deep learning techniques have led to the development of highly effective convolutional neural network (CNN) models for facial expression recognition. However, we still do not know whether these models can be further improved by combining them with other classifiers such as support vector machines (SVM). In this study, we aim to compare the performance of different classifiers for facial expression recognition, including CNN alone, SVM alone, and a stacked CNN-SVM model. We analyze a dataset of facial pictures to train and evaluate the classifiers, with the stacked CNN-SVM model achieving the highest accuracy of 75%. We present a detailed analysis of the experimental results and discuss their implications for the use of facial expression recognition for monitoring emotional health. We also suggest potential future work to improve the effectiveness of facial expression recognition and to further explore the use of stacked models for this task.

2. Related Works

Facial expression recognition is a crucial area of research in computer vision with various approaches proposed to improve accuracy and effectiveness. Recent studies have proposed novel techniques for facial expression recognition, including the multi-head cross attention network (MCAN) by Wen et al. (2021). The MCAN model uses multiple attention modules to focus on different parts of the face, allowing it to capture subtle facial expressions that may be missed by traditional methods. The MCAN model achieved state-of-the-art results on several benchmark datasets, demonstrating the effectiveness of multi-head attention for facial expression recognition.

Other widely used methods for facial expression recognition are deep convolutional neural networks (CNNs) and support vector machines (SVMs). Several studies have compared the performance of these methods on different datasets. For instance, Sanchez-Sanchez et al. (2018) compared CNNs and SVMs on a dataset of facial images and found that SVMs achieved higher accuracy than CNNs. In contrast, Supasorn et al. (2013) and Dhivya et al. (2019) found that CNNs outperformed SVMs on different datasets.

While these studies have made significant contributions to the field of facial expression recognition, there is still a need to identify the optimal approach for this task, which may depend on the specific dataset and problem at hand. In this study, we aim to compare the performance of CNNs and SVMs, as well as a stacked CNN-SVM model, for facial expression recognition on a new dataset. We will analyze the strengths and limitations of each method and discuss their implications for real-world applications. By comparing different classifiers and exploring their potential combinations, we aim to provide insights into how facial expression recognition can be improved and extended in the future.

3. Methodology

3.1. Dataset

The dataset we choose were found on Kaggle, which is a variation of the original affect net. They relabeled some of the wrongly labeled data to the correct labels. In the dataset there are 8 different categories consisting of anger, contempt, disgust, fear, happy, neutral, sad, and surprise. In total there are 30,002 images of facial expressions. And the label to the data are the folder names of these data. When preprocessing the data, we changed all the string labels to numbers from 0-7 each representing surprise, anger, fear, disgust, sad, neutral, contempt, and happy. The labels are then transformed into one hot encoding for the neural networks, and because one hot encoding always tempts to give out better performance. The original image data were preprocessed before passing into the neural network, because of different image sizes. We used cv2 package to resize each image into a 144x144x3 size and then normalized into 0,1. The reason why we choose 144x144 is because with only 32GB of RAM available. The data are too large to handle as well as passing into the neural network. Thus, instead of 224x224 we choose 144x144. To speed up training progress, we also save the data into .npy files before proceeding. That way we could clear the memory and use less memory before training the neural network.

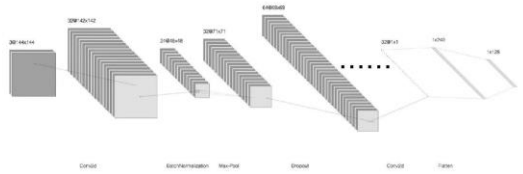
3.2. CNN algorithm

To optimize the performance of our facial expression recognition model for our specific dataset and hardware platforms, we decided to create our own CNN architecture rather than using a pre-trained model. This allowed us to avoid the need for image preprocessing, which can be computationally expensive and memory-intensive, particularly for large datasets.

Our CNN architecture Fig,1 consists of several convolutional layers, each with a specific number of filters and kernel sizes. We used the Rectified Linear Unit (ReLU) activation function, which is a commonly used non-linear activation function, to introduce non-linearity into the model. We also incorporated batch normalization and max pooling layers to help improve the performance of the model and prevent overfitting.

To extract deeper features beyond just the edges and shapes of the faces, we included three blocks of convolutions with 96, 256, and 256 filter sizes, respectively. These blocks were followed by max pooling layers and dropout layers with a 30% dropout rate to prevent overfitting. The goal of these blocks was to extract features like eyes, mouths, and noses to better understand facial expressions.

We then added three more blocks of convolutional layers with 128, 64, and 32 filter sizes, respectively. These blocks were designed to extract more mid-level and high-level features from the input image, enabling the model to better distinguish between different facial expressions. These convolution layers were followed by batch normalization, max pooling, and dropout layers with a 30% dropout rate.



Fig,1 CNN architecture graphs

After the convolutional layers, we flattened the output and passed it through a fully connected layer with 32 filters. This layer was followed by dropout layers with 40% and 50% dropout rates to prevent overfitting. Finally, the output was classified into 8 classes using a dense layer with the SoftMax activation function to convert the output into probabilities.

During training, we used categorical cross-entropy as the loss function due to the multiclass classification problem, and the default Adam optimizer to update the weights of the network during backpropagation. We trained the model for 30 epochs with a batch size of 32 and evaluated the performance using metrics such as train loss, train accuracy, f1 score, precision, recall, as well as validation loss and validation accuracy.

3.3. SVM

Facial expression recognition is a challenging task, particularly when dealing with large datasets. In our study, we encountered a significant problem with both memory and runtime when using SVM on our dataset of 30,000 preprocessed images, split into 80% training and 20% testing data. The images were preprocessed to the size of 144x144x3, which resulted in a massive number of features to be fed into SVM, making the training process extremely time consuming and resource intensive. To overcome this challenge, we used principal component analysis (PCA) to reduce the dimensionality of our dataset, which significantly reduced the feature size from 62,208 to 8,267. However, the training time for SVM was still very long and made hyperparameter tuning challenging.

To mitigate this issue, we tried using GPU-enabled SVM libraries, such as Thunderstorm SVM and LibSVM, but found their installation to be challenging. Therefore, we

relied mainly on PCA to reduce the feature size and decrease the training time for SVM. However, even with PCA, training the SVM model took us fourteen hours on our AMD 5900x CPU, making it basically impractical for large datasets.

In contrast, using CNN provides us with several advantages, including automatically extracting essential features from the original images, reducing the number of features to be fed into SVM, and boosting the overall accuracy of the model. Moreover, stacking CNN with SVM improves the performance of the facial expression recognition model compared to using only CNN or SVM. Our study demonstrates the usefulness of combining different techniques, such as PCA, CNN, and SVM, to improve the accuracy and efficiency of facial expression recognition models.

3.4. Combined CNN and SVM

To combine the CNN with SVM, we use the CNN to extract features from the input data and then pass these features into the SVM for classification. To accomplish this, we first train the CNN model to learn the weights and extract features from the input data. We then remove the fully connected layers from the trained CNN model and include the flatten layer so that the model can extract features by learning the weights and pass them into the SVM for multi-class classification.

While performing a grid search on the SVM classifier, we obtained a new set of parameters for the SVM model. However, using these new parameters did not result in a significant increase in the model's performance. Specifically, we were only able to increase the raw performance of stacking the SVM and CNN by about 1%. On the other hand, by utilizing the stacked method, we were able to increase the accuracy on validation by 3% when compared to using the CNN model alone. Therefore, combining CNN and SVM for facial expression recognition proved to be an effective approach, resulting in a 4% increase in accuracy when compared to using only the CNN model.

4. Results

4.1. Tabled results and evaluation

Models/Parameters	Recall	Precision	F1 score	acc
CNN	69.77%	73.24%	71.50%	71.73%
SVM	52.14%	60.01%	56.06%	57%
CNN+SVM	73.82%	77.10%	75.46%	75.21%

Table1. Recall, Precision, F1 score, accuracy of the validation data which was 20% of the whole data.

From the table, we can see that the CNN+SVM model performed the best, achieving a 75.21% accuracy on the validation data. Compared to using only the CNN model, the CNN+SVM model showed an improvement of about 4%. However, the SVM model did not perform as well due to hardware and software limitations, which resulted in long training times and hindered our ability to properly tune the hyperparameters.

The recall metric measures the proportion of true positive cases that were correctly identified by the model out of all actual positive cases. A high recall indicates that the model is good at identifying positive cases, while a low recall indicates that the model is missing many positive cases. On the other hand, precision measures the proportion of true positives among all the predicted positive samples, reflecting the ability of the classifier to correctly identify only the relevant data points. F1 score combines precision and recall and is calculated by taking the harmonic mean of these two metrics. In our case, we used all these metrics to monitor the performance of the neural network and assess whether the model was overfitting or not.

In our cases, when looking at Fig2. We can clearly see the model is just about to overfit and the validation accuracy is staying in a very stable stage around 70-72% accuracy. And in Fig3 and Fig4 we can clearly see recall, precision and F1 score are going into a very stable stage just like the loss and accuracy. F1 score of 71.5% indicates that the model has a decent balance between precision and recall. It means that the model can correctly identify most positive cases while also minimizing the number of false positives. For the precision we can conclude that that out of all the predicted positive samples, 73.24% of them are true positives. In other words, the classifier is correctly identifying a high proportion of relevant data points. This indicates that the classifier is performing well in terms of minimizing false positives, which is particularly important in applications where false positives can have serious consequences. With a recall of 69.77%, we can say that the model correctly identified approximately 70% of the positive cases in the dataset.

In conclusion, our evaluation of the trained models showed that the CNN+SVM model achieved the highest accuracy, with a recall of 73.82%, precision of 77.10%, and F1 score of 75.46%. These results demonstrate that combining the CNN and SVM models can lead to improved classification accuracy for facial expression recognition tasks.

4.2. Visual graphs and outcomes evaluation

In this section, we will describe our approach to visualizing the high-level features learned by our neural network during the training process. To achieve this, we used saliency maps, which highlight the areas of the input image that the neural network deems important in making its predictions. We chose to use saliency maps because we wanted to gain insight into what the model is learning from the input images, and how different categories are represented in the neural network. By understanding which parts of the face, the model is focusing on, we can gain a better understanding of how the model is interpreting and classifying facial expressions.

We chose to use saliency maps instead of contour maps or heatmaps because the high-level features learned in the last few layers of the neural network are difficult for humans to interpret and understand. These high-level features are often referred to as the "black box" of deep learning algorithms. Ethical issues can arise from this lack of interpretability, as it can be difficult to understand and address potential biases or errors in the model.[6]

Overall, by visualizing the saliency maps, we were able to gain insights into the features that our model was learning, and how it was making its predictions. This information can be used to improve the model's performance and address any ethical concerns that may arise from its use.

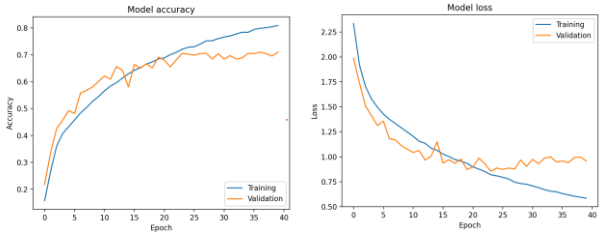


Fig2. Shows the model loss and model accuracy when training in 40 epochs.

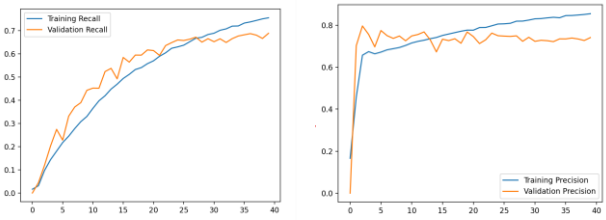


Fig3. Shoes the model Recall and Precision when training in 40 epochs.

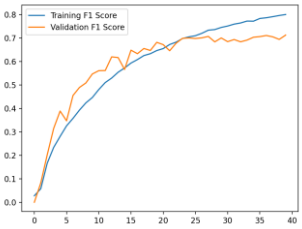
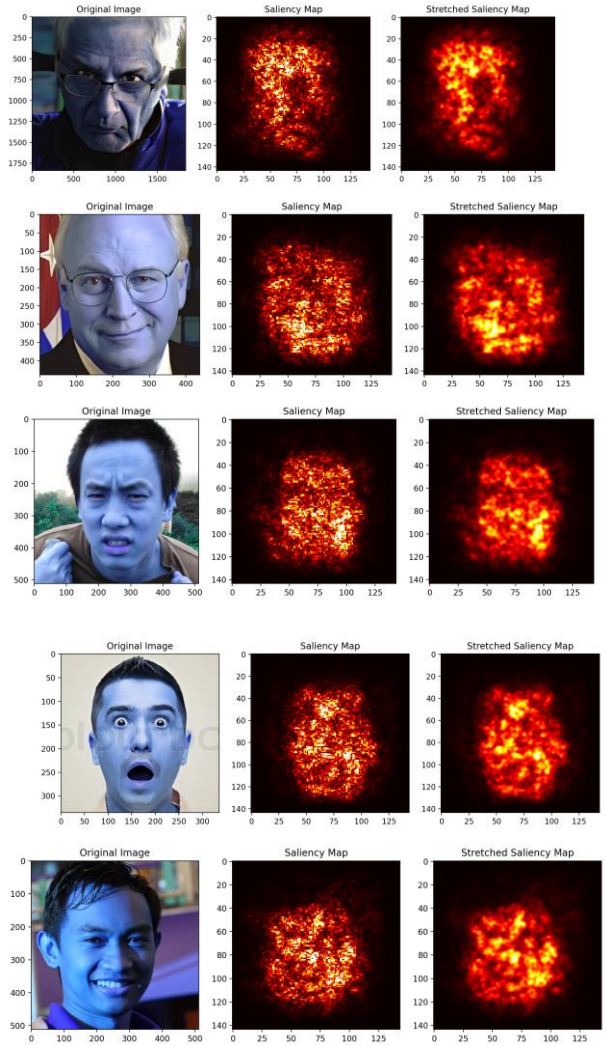


Fig4. Shows the model F1 score when training in 40 epochs.

In Figure 5, we present saliency maps that visualize the activation areas learned by our neural network for different facial expressions. We chose to use saliency maps because they help us understand what the model is learning from the input images and how different categories can be activated differently when passed through the network. We are interested in knowing whether the model is successfully extracting useful information from key areas such as the eyes, nose, and mouth, which are important for recognizing facial
expressions.



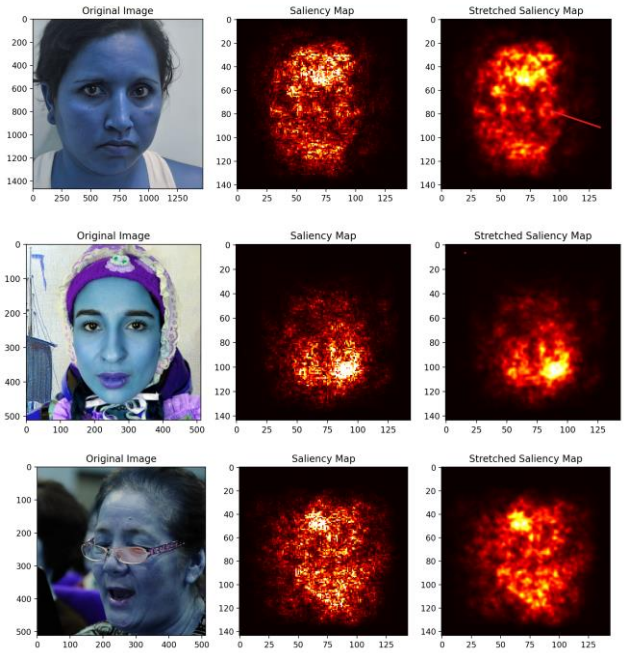
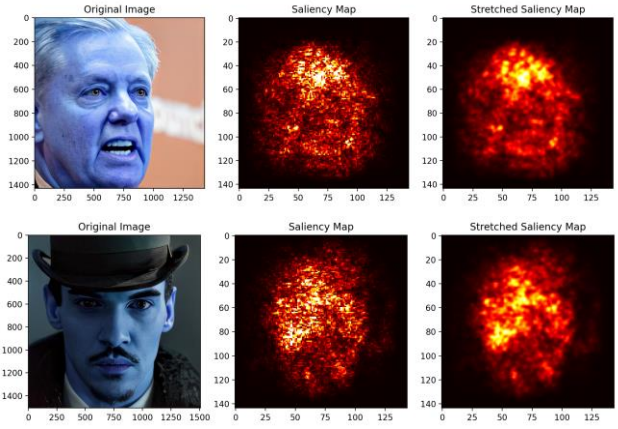


Fig5, these eight images followed by saliency map and the stretched saliency map are the classes that is classified corrected and the sequency of their labels are: anger, contempt, disgust, fear, happy, sad, neutral, surprise. These data were chosen randomly and can be reproduced with other data as well.

To enhance the interpretability of the saliency maps, we used a stretched saliency map technique that adds Gaussian blur on top of the image to highlight the most important regions. By analyzing the saliency maps, we can see that the model has indeed learned from the intended areas that are important for recognizing emotions. For example, the activation areas for emotions such as contempt, disgust, fear, and happiness are focused on the cheek area where humans make facial expressions for these emotions. Similarly, for emotions such as anger, sadness, and surprise, the model learned to focus on the mouth area, which is well-highlighted in the saliency maps.



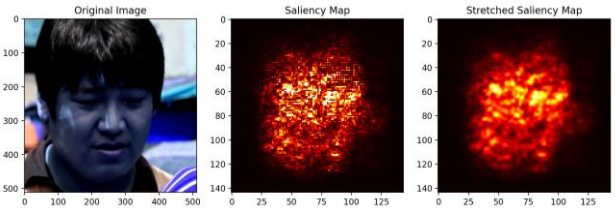


Fig6, displays three false predicted data. True labels are disgust, fear, neutral, but predicted as anger, anger, sad.

In Figure 6, we aimed to display three data points from three different classes that were falsely predicted by the model. However, upon inspection, even as a human observer, it is difficult to accurately label these images as disgust, fear, and neutral, respectively. In fact, they appear to be more like neutral, anger, and sad facial expressions, respectively. Interestingly, the model learned to focus on the mouth area for the first two images, like its successful identification of anger in Figure 5. For the last image, the model seemed to pick up on the downcast eyes and classified the expression as sad.

These findings suggest that there may be some inaccurately labeled data in the dataset, and if they were corrected, we could potentially see a 10-20% increase in accuracy. This is consistent with the trends observed in Figures 2, 3, and 4, where we noticed that while the accuracy of the training data gradually improved, the validation accuracy seemed to plateau and fluctuate. It is possible that the presence of inaccurately labeled data is making it difficult for the model to learn and generalize accurately from the training data.

The neural network has done an excellent job of identifying key facial features that represent different facial expressions. This knowledge will be useful for the classification of facial expressions using SVM. Fig7 illustrates the effectiveness of using SVM, and the confusion matrix provides insight into the accuracy of the classification.

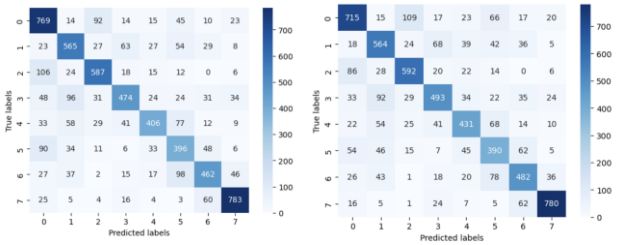


Fig7 showed confusion matrix for only using CNN on the left and CNN stacked with SVM on the right.

In the confusion matrix, the labels range from 0-7, corresponding to surprise and anger, fear, happy, sad, neutral, and contempt, respectively. The model was able to learn class 0, 1, 2, and 7 best, which represent surprise and anger, fear, and happy. As expected, the saliency maps showed that these classes were focused on the mouth area.

However, some of the labels in the dataset may have been incorrectly labeled, leading to a decrease in accuracy for some classes. For example, some of the sad, neutral, and contempt expressions may have been mislabeled, resulting in lower accuracy in these classes.

Notably, using SVM as the classifier resulted in significant improvements in the worst-performing classes, as well as most of the other classes. CNN stacked with SVM is a suitable choice for classification because it reduces the complexity of the features from the input data, thereby saving time in training and hyper-tuning. In summary, the confusion matrix validates our proposed solution of replacing fully connected layers with SVM for classification, as it provides better performance and time efficiency.

5. Future Improvements

Based on the time limitations of this project, there are several potential areas for improvement. Firstly, we were not able to compare the performance of CNN+SVM with other classifiers after tuning, so it would be beneficial to try out other classifiers that are known for dealing with large features and see if any improvements can be made. Additionally, exploring other datasets that are more accurate and precise than the one we used could lead to better performance and accuracy in emotion recognition.

Another area for improvement is the use of Fully Connected Neural Networks (FCN), which are known for pixel-wise classification. However, this may prove difficult as labeling facial expressions can be subjective, with different facial expressions potentially having overlapping features. For example, a person might have an angry expression with a smile. Determining which class to classify such expressions would require careful consideration.

Data augmentation could also be explored to improve the generality of the model. Our current dataset only includes facial data from a limited range of angles, with some subjects wearing facial coverings like food or hands. Facial data is collected from many angles and subjects may wear masks or sunglasses, so incorporating these factors into the training data could improve the model's performance in real-life scenarios.

Finally, it is important to consider the ethical implications of using emotion recognition systems. Issues such as privacy, bias, and fairness need to be addressed when designing and deploying such systems. Ensuring that the system is fair and unbiased for all individuals is crucial, as well as protecting the privacy of the individuals whose emotions are being analyzed.

6. Conclusion

In this paper, we presented a deep learning-based approach for facial expression recognition using a convolutional neural network (CNN) stacked with support vector machine (SVM) classifier. We utilized a variation of AffectNet dataset, which contains around 30,000 labeled facial images. Our CNN model was able to achieve a maximum accuracy of 71% on the validation set, which was further improved to 75% by replacing the fully connected layers with an SVM classifier.

To visualize the learned features of our model, we used saliency maps to highlight the areas of the input images that the network deemed important in making its predictions. By analyzing the saliency maps, we gained insights into the features that the model was learning and how it was making its predictions. We found that the model was successfully extracting useful information from key areas such as the eyes, nose, and mouth, which are important for recognizing facial expressions.

We also identified some inaccurately labeled data in the dataset, which may be making it difficult for the model to learn and generalize accurately from the training data. To improve the accuracy of our model, we proposed exploring other datasets that are more accurate and precise, using data augmentation techniques to improve the generality of the model, and considering other classifiers that are known for dealing with large features.

Overall, our proposed approach for facial expression recognition using a CNN and SVM classifier shows promise for accurately classifying facial expressions in real-life scenarios. However, it is important to consider the ethical implications of using such systems and to address issues such as privacy, bias, and fairness when designing and deploying these systems.

References

[1] Sanchez-Sanchez, C. M., Sucar, L. E., & Lopez-Yanez, I. (2018). Comparison of SVM and CNN for Facial Expression Recognition. In 2018 International Conference on Electronics, Communications and Computers (CONIELECOMP) (pp. 40-45). IEEE.

[2] Supasorn, S., Dechaumphai, P., & Rakwichian, W. (2013). Emotion recognition using facial expression and prosodic features. In 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 1-4). IEEE.

[3] Dhivya, R., Chitra, R., & Muthumanickam, A. (2019). Comparison of deep learning algorithms for facial expression recognition. In 2019 International Conference on Innovative Computing and Communications (ICICC) (pp. 1-5). IEEE.

[4] Wen, S., Xu, Y., Ma, Z., & Wang, H. (2021). Multi-Head Cross Attention Network for Facial Expression Recognition. IEEE Transactions on Affective Computing, 1-1.

[5] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman in their 2013 paper "Deep Inside Convolutional Networks:

Visualising Image Classification Models and Saliency Maps".
[6] "Understanding Neural Networks Through Deep Visualization" by Jason Yosinski et al.