Youssif Ashraf Abuzied 900202802
Sara Maged Naseef 900202369

Advanced Machine Deep Learning

Facial Expression Recognition

26th September 2023

Problem Statement and Motivation.

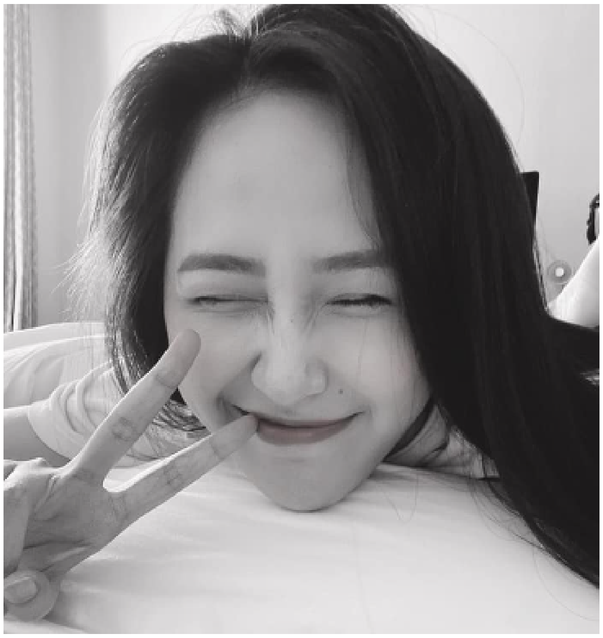
Computer vision is one of the fastest growing fields in machine learning. It opens the gates for computers to interpret and understand visual data to use them in many useful applications. One of the most exciting topics in computer vision is facial expressions recognition, commonly referred to as (FER). Facial expression recognition is a computer vision task which aims to identify and categorize emotional expressions that are depicted on a human face. The goal behind this task is to make the process of determining in real time automated, by analyzing various features in a human face such as eyes, mouths, eyebrows, noses, and many other features and trying to find correlations between those features and many human emotions such as happiness, anger, sadness, disgust, surprise, etc.

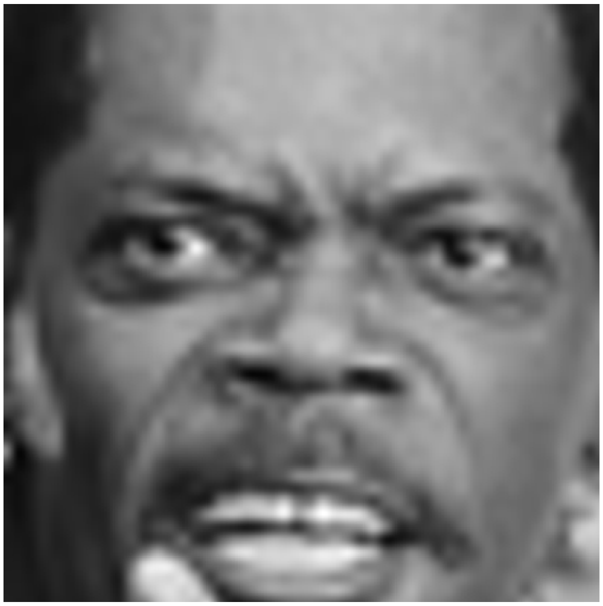
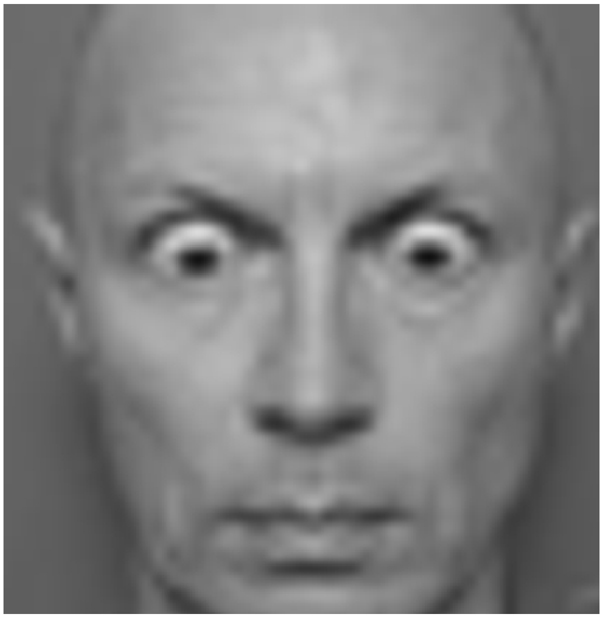
This task has a variety of applications. For instance, in health care applications, facial expression recognition can help doctors better monitor their patients and better understand their psychological states. Moreover, FER is of extreme usefulness in security and surveillance systems as they can use FER for tasks like automated airport screening. Additionally, User experience researchers can use FER to determine how users respond to different designs or stimuli. Finally, interactive devices like smartphones and virtual assistants aim to provide more human-like and contextual responses by understanding user emotions conveyed through facial cues.

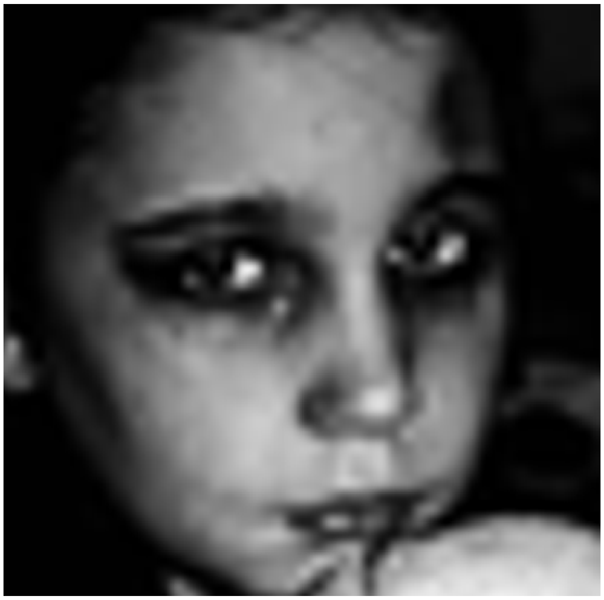
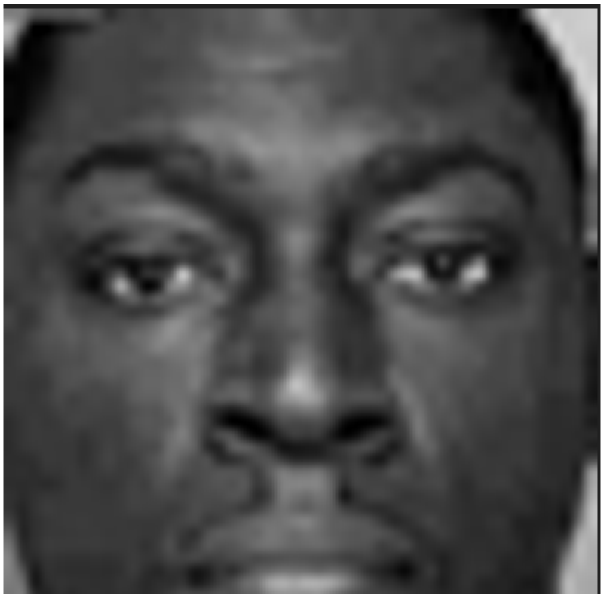
This topic has gained a lot of interest from researchers over the years. Many models have been developed to handle this task and to optimize its results. In our research, we intend to explore the different approaches and state of the art models that have been developed to tackle this problem. Also, we will survey the different datasets that could be used in this problem. Finally, we will propose and apply some updates to some of the existing models in order to optimize their performances.

Input / Output.

Below are some examples of the inputs to the problem and their respective outputs.

 fig.1	Happy

 fig.2	Angry
 fig.3	Surprised

 fig.4	Sad
 fig.5	Neutral

Survey of the available evaluation metrics or tools for this problem.

Our problem is a multi label classification problem which aims to recognize the emotion the person in the image feels. The following metrics are the most widely used metrics in the literature for this type of problem.

First, Accuracy.

Accuracy is by far the most widely used metric in classification problems, it basically calculates the number of correct instances the model has predicted. It can be calculated using the following formula

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Although accuracy is by far the most used metric and can tell us about the effectiveness of a model to a great extent, some downsides exist for the accuracy. For instance, consider the case when we have a model that predicts whether a person has covid or not, and it is known that in 95 percent of the cases, the person is not covid carrier. Assume that our model always predicts that the person is not a covid carrier. We will end up with 95 percent accuracy which is good. However, the model is not capable of predicting any person who has covid. As a result, depending only on the accuracy can lead to disastrous results. That is why a lot of other metrics have been developed to determine the effectiveness of the model with respect to a given class of output, not the overall output. Before discussing those metrics, we need first to introduce some terms that will be used repeatedly in this proposal.

True Positive: Means that the model predicted a positive result for a given class (ex. This person is happy) and the truth value is positive.

False Positive: Means that the model predicted a positive result for a given class (ex. This person is happy) and the truth value is negative.

False Negative: Means that the model predicted a negative result for a given class (ex. This person is not happy) and the truth value is positive.

True Negative: Means that the model predicted a negative result for a given class (ex. This person is not happy) and the truth value is positive.

Second, Precision.

Precision measures the fraction of the positive results which are actually positive.

It can be measured with the following formula:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Note that this formula calculates the precision of a specific class. If we want to generalize it, we can take the average of the precisions of all the classes. Another approach is to take the weighted average of all the precisions. The use case of the model determines which of the approaches to use.

Third, Recall.

The recall metric calculates that number of the positive classes that the model was able to detect. It can be calculated with the following formula:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

Considering the covid model that was discussed before, the recall metric will be of great usefulness. Our aim in the covid model is to be able to detect people who are covid carriers with minimal error. The recall metric helps in this case as it is able to calculate the fraction of covid carriers that the model successfully detected. If we want to generalize this metric to all classes, the two approaches discussed in the precision can be used here too.

Fourth, F1 score.

F1 score is a way of combining precision and accuracy into a single metric. It can be calculated using this formula:

$$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Fifth, the ROC curve and AUC.

ROC curve refers to the Receiver Operating Characteristic which is a 2 dimensional curve that has False Positive Rate on the horizontal axis and the True Positive Rate on the vertical axis. We can construct this plot by calculating the true positive rate and the false positive rate at different model thresholds or parameters and then plot those values.

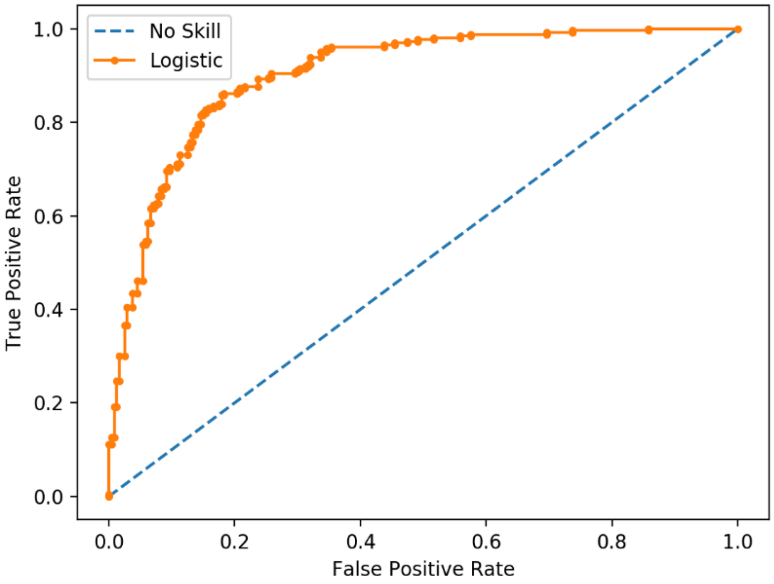
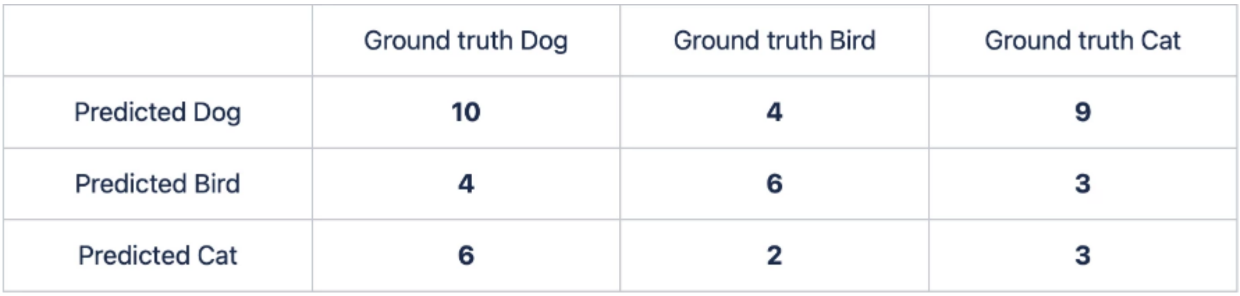


fig.6

AUC refers to the area under the curve of the ROC curve when the area gets Larger. his means that the model is doing well.

Sixth, the confusion matrix.

A confusion matrix is a way to understand how the model is behaving across different classes. In the confusion matrix, the number of correct and incorrect predictions are summarized with count values and broken down by each class. Below is an example of a confusion matrix.



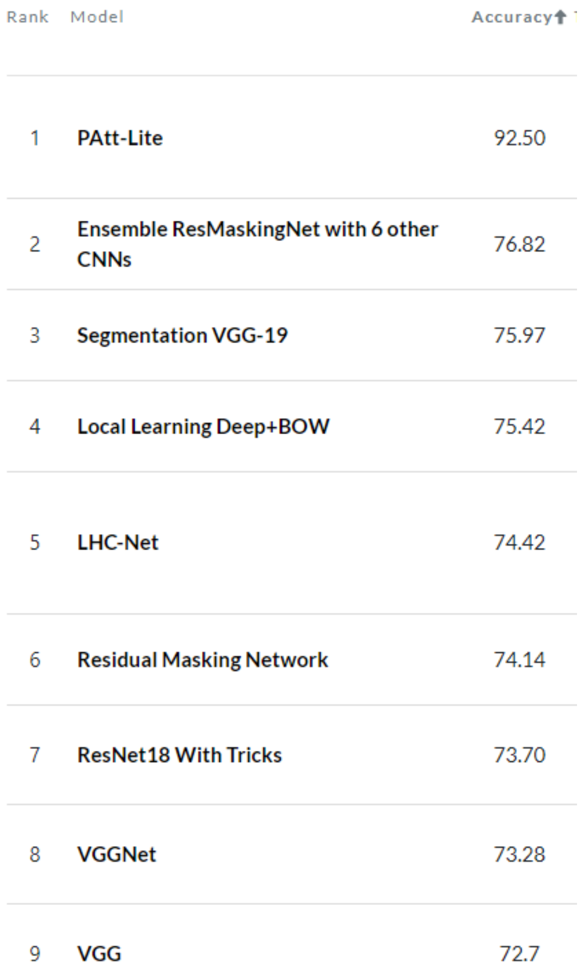
	Ground truth Dog	Ground truth Bird	Ground truth Cat
Predicted Dog	**10**	**4**	**9**
Predicted Bird	**4**	**6**	**3**
Predicted Cat	**6**	**2**	**3**

fig.7

Note that when the values on the diagonal starting at the upper left corner gets larger this means that the model is doing well.

Current state of the art results.

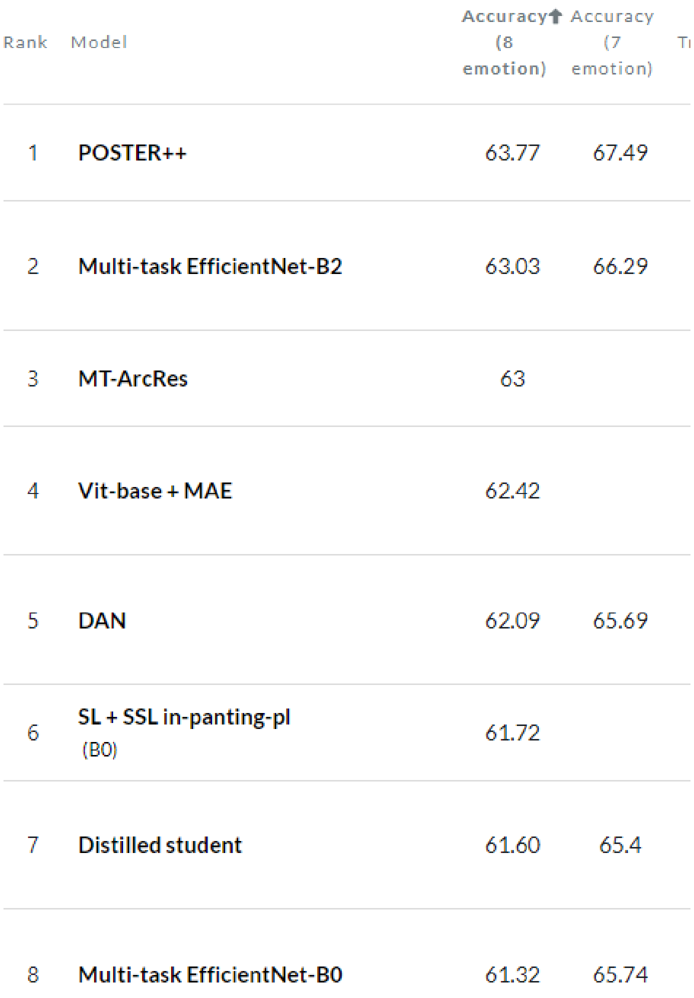
The following table shows the state of the art results of Facial Expression Recognition on the FER 2013 Dataset.



Rank	Model	Accuracy↑
1	PAtt-Lite	92.50
2	Ensemble ResMaskingNet with 6 other CNNs	76.82
3	Segmentation VGG-19	75.97
4	Local Learning Deep+BOW	75.42
5	LHC-Net	74.42
6	Residual Masking Network	74.14
7	ResNet18 With Tricks	73.70
8	VGGNet	73.28
9	VGG	72.7

fig.8

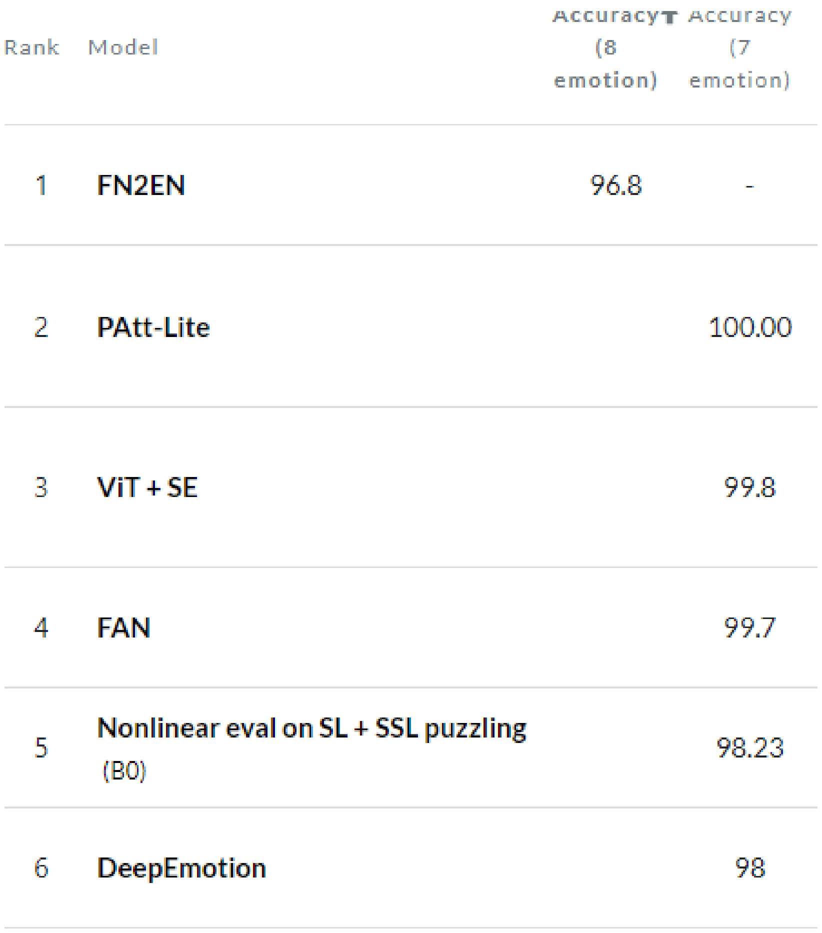
The following table shows the state of the art results of Facial Expression Recognition on the AffectNET Dataset.



Rank	Model	Accuracy↑ (8 emotion)	Accuracy (7 emotion)	Ti
1	**POSTER++**	63.77	67.49	
2	**Multi-task EfficientNet-B2**	63.03	66.29	
3	**MT-ArcRes**	63		
4	**Vit-base + MAE**	62.42		
5	**DAN**	62.09	65.69	
6	**SL + SSL in-panting-pl** (B0)	61.72		
7	**Distilled student**	61.60	65.4	
8	**Multi-task EfficientNet-B0**	61.32	65.74	

fig.9

The following table shows the state of the art results of Facial Expression Recognition on the CK+ Dataset.



Rank	Model	Accuracy (8 emotion)	Accuracy (7 emotion)
1	**FN2EN**	96.8	-
2	**PAtt-Lite**		100.00
3	**ViT + SE**		99.8
4	**FAN**		99.7
5	**Nonlinear eval on SL + SSL puzzling** (B0)		98.23
6	**DeepEmotion**		98

fig.10

Survey of the available datasets.

First, FER2013 Dataset.

Dataset Description: This dataset consists of 48x48 grayscale facial images. This dataset was released in 2013 in a competition for facial image recognition hosted by kaggle. This dataset has 34034 images splitted into two different sets. A training set which contains 28709 images, a testing set which has the rest of the images. Each image in this dataset is assigned 1 of 7 different labels which are Angry, Disgust, Happy, Fear, Sad, Neutral, and Surprise. The images in this dataset have been preprocessed in a way such that all the faces are centered in the image.

Dataset Size: Around 700 MBs.

Dataset Link:
https://www.kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge/data

Examples from the dataset:

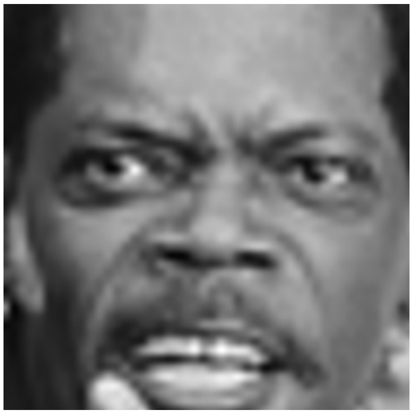
Image	Label
 fig.11	Angry

 fig.12	Happy

Second, AffectNet Dataset.

Dataset Description: This dataset consists of 96x96 RGB images that are used for Facial Expression Recognition. This dataset contains about 28.6 images. There is no mention of any training or testing splits in this dataset. Each of the images is assigned one label out of 8 which are ((neutral, happy, angry, sad, fear, surprise, disgust, contempt) .

Dataset Official Link:

http://mohammadmahoor.com/affectnet/

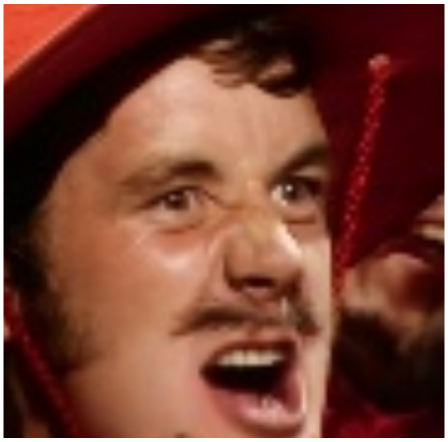
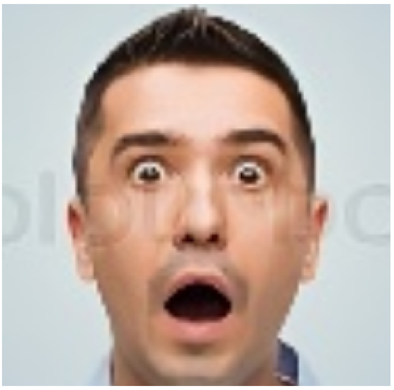
Dataset Download Link:
https://www.kaggle.com/datasets/noamsegal/affectnet-training-data?select=labels.csv

Dataset Size: About 350 MBs.

Side Note: We do not have access to the official dataset yet and we submitted a request to access it. However, the part of this dataset we found at kaggle is very enough for the scope of our project. Also, this dataset is not our first choice.

Examples from the dataset:

Image	Label

 fig.13	Anger
 fig.14	Fear

Third, FER+ Dataset.

Dataset Description: This dataset contains exactly the same image in the FER 2013 dataset. The only difference is that this dataset has more refined labels than the FER 2013 dataset. Another difference is that this dataset has an additional label that was not included in the FER2013 dataset which is Contempt.

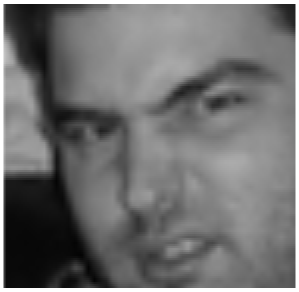
Dataset Official Link:

https://github.com/Microsoft/FERPlus

Dataset Size: Around 700 MBs.

Examples from the dataset:

Image	**Label**

 fig.15	**Disgust**
 fig.16	**Neutral**

Third, CK+ Dataset.

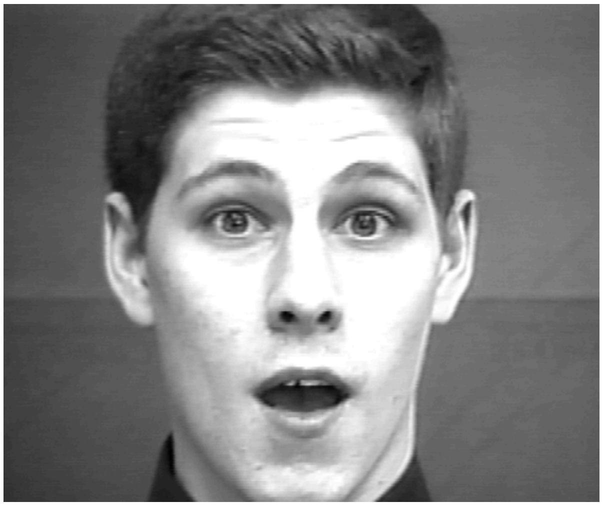
Dataset Description:

The Extended Cohn-Kanade (CK+) dataset consists of 593 video clips from 123 distinct people who range in age from 18 to 50 and are of various genders and ethnic backgrounds. Each video depicts a change in face expression from neutral to a specified peak expression. It was shot at 30 frames per second (FPS) at either 640x490 or 640x480 resolution. One of the seven expression classes—anger, contempt, disgust, fear, pleasure, sorrow, and surprise—is assigned to 327 of these movies. The majority of facial expression classification algorithms employ the CK+ database, which is widely recognized as the most frequently used laboratory-controlled facial expression classification database currently available.

Dataset Link: **https://www.kaggle.com/datasets/stefanaduta/ckdataset**

Dataset Size: Around 2 GBs.

Image	**Label**

 fig.17	**Surprised**

Selected Dataset.

After careful consideration of the different aspects related to the dataset, we decided to go on with the FER + and theFER 2013 datasets for the following reasons:

1. The number of images in these dataset is very good and we will be able to build a good model based on it.
2. The size of these datasets is not large. As a result, training will not take a huge time and we will not face memory issues. Also, because of its good size, we can run the model on some online platforms such as google colab for training and testing.
3. Most of the models in the field of Facial Expression Recognition were trained and reported results on this dataset. As a result, we will be able to test the performance of our final model against several models.
4. This dataset is preprocessed in its nature. In other words, some preprocessing has been done on this dataset to make the faces in the center.
5. We will use these two datasets together, not one of them because each of them will help us in a different task. First, the FER2013 dataset contains a lot of noisy labels and one of the tasks that we will be doing is to make our model able handle noisy labels and not to overfit them. For general training, we will be using the FER+ dataset as its labels are more accurate and refined.

Survey of the available models.

Unlike most of the proposed solutions for this problem that focuses entirely on CNN and different architectures for it, we intended to approach it from the attention based machine learning mechanism. As a result, all of the surveyed models approached this problem from the angle we intended.

First, EAC.

Introduction to the model:

Erasing attention consistency (EAC) model was introduced in the paper: Learn From All: Erasing Attention Consistency for Noisy Label Facial Expression Recognition. In this paper, the researchers' main focus was about developing an efficient mechanism to deal with the noisy labels in the dataset. They stated in facial expression recognition tasks, there are a lot of noisy labels due to the intra class similarity and the annotation ambiguity. To address this problem, they did the following. First, they developed a mechanism to suppress noisy labels during the training process immediately. Second, they utilized the fact that flipping an image has no semantic difference to design an imbalanced framework. Then, they randomly erased parts of the images. Next, they used flip attention consistency to prevent the model from focusing on some parts of the features only. Finally, they realized that the EAC model they built outperformed the state of the art models that did not develop a mechanism to handle the noisy labels.

Architecture:

The whole architecture of the EAC model is depicted in the figure below. In short, the model works as follows. It gets a batch of images and then erases parts of the images in a random way. The set of randomly erased images is denoted as **I**.Then, the model flips the I images, to get a set of flipped images called **I'**. Then, both **I** and **I'** enter a backbone model (they experimented with a backbone model which is ResNet-18. Next, the model gets the feature maps of both **I** and **I'** from the last layer of the backbone model which are denoted **F** and **F'** respectively. Then, the model sends only F to a GAP (Global average Pooling) to get the features **f** which has the dimension of N x C where N is the number of images and C is the number of channels. Now, features f are sent to a FC (Fully connected layer) to compute the classification loss according to the formula:

$$l_{cls} = -\frac{1}{N}\sum_{i=1}^{N}(\log \frac{e^{\mathbf{W}_{\mathbf{y}_i}\mathbf{f}_i}}{\sum_j^L e^{\mathbf{W}_j\mathbf{f}_i}}),$$

In the above formula Wyi denotes the weight of yi image from the FC layer. It is worth noting that classification loss is calculated using the feature map F only. Now, the attention maps **M** and **M'** are calculated using the formula:

$$\mathbf{M}_j(h, w) = \sum_{c=1}^{C} \mathbf{W}(j, c)\mathbf{F}_c(h, w),$$

Finally, the consistency loss between M, and M' is calculated using the formula:

$$l_c = \frac{1}{NLHW}\sum_{i=1}^{N}\sum_{j=1}^{L}||\mathbf{M}_{ij} - Flip(\mathbf{M}^{'})_{ij}||_2.$$

The total loss is then calculated using the formula:

$$l_{total} = l_{cls} + \lambda l_c.$$

Note that y is a hyperparameter of the erasing consistency loss.

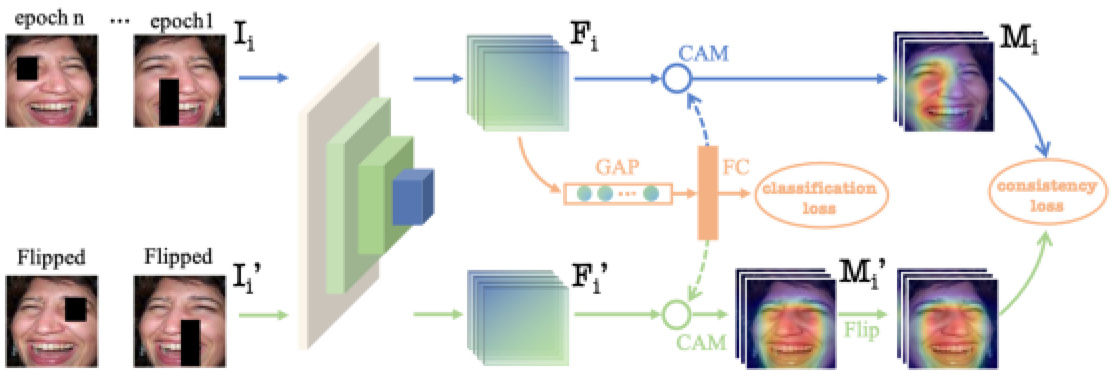


fig.18

Paper Link: Learn From All: Erasing Attention Consistency for Noisy Label Facial Expression Recognition

Datasets:

Researchers trained and tested their model on different datasets including: FERPlus, RAF-DB, and AffectNet.

Model Source Code:

https://github.com/zyh-uaiaaaa/erasing-attention-consistency

Model Weights:

The repository of the code includes only a pre-trained version of the backbone Model but no weights are found for the whole model.

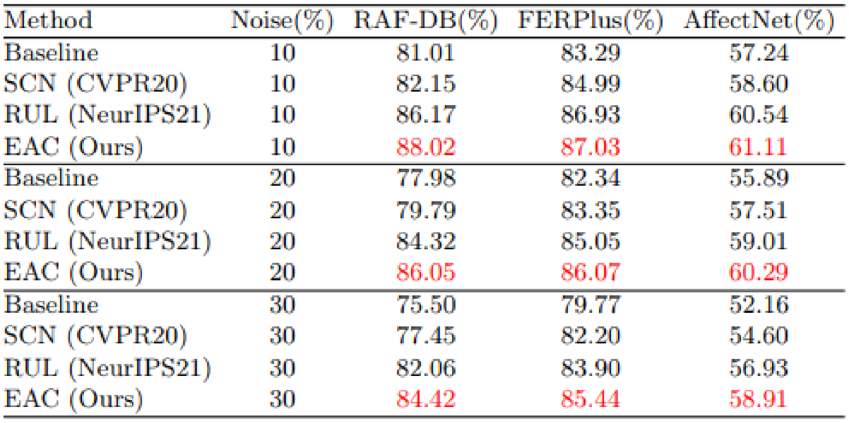
Information about the frameworks used in the code:

It was mentioned in the github repo that the model was built using Torch 1.8.0 and torchvision 0.9.0.

Information about the training resources: Not Mentioned.

Results:

It is clear from the table that the model overachieved some of the state of the art models on different datasets.



Method	Noise(%)	RAF-DB(%)	FERPlus(%)	AffectNet(%)
Baseline	10	81.01	83.29	57.24
SCN (CVPR20)	10	82.15	84.99	58.60
RUL (NeurIPS21)	10	86.17	86.93	60.54
EAC (Ours)	10	88.02	87.03	61.11
Baseline	20	77.98	82.34	55.89
SCN (CVPR20)	20	79.79	83.35	57.51
RUL (NeurIPS21)	20	84.32	85.05	59.01
EAC (Ours)	20	86.05	86.07	60.29
Baseline	30	75.50	79.77	52.16
SCN (CVPR20)	30	77.45	82.20	54.60
RUL (NeurIPS21)	30	82.06	83.90	56.93
EAC (Ours)	30	84.42	85.44	58.91

fig.19

Second, FER-VT.

Introduction to the model:

The Facial Expression Recognition model with Visual Transformers (FER-VT) was introduced in the paper: Facial expression recognition with grid-wise attention and visual transformer. The paper starts by describing the problem of facial expression recognition (FER) and introduces the FER-VT model as a solution to improve FER performance. It mentions three widely used datasets for FER: CK+, FER+, and RAF-DB, which contain labeled facial expressions. The paper then suggests a novel framework with two attention mechanisms (which is FER-VT) for CNN-based FER models at both the low-level feature learning stage and high semantic representation stage.. The two proposed mechanisms overcome the weakness of CNN-based models in learning long-range inductive biases for enhancing the performance of performing a FER task. The designed model uses a grid-wise attention mechanism that uses long-range dependencies between different facial regions to regularize the convolutional parameter learning in the low-level feature extraction for a FER task. The model also proposes a token-based visual transformer to learn long-range inductive biases in high-level semantic feature learning. Also, the results provided in this paper demonstrate that FER-VT has achieved state-of-the-art performances.

Architecture:

The model framework is divided into three parts: Backbone Network, Grid-Wise Attention (GWA), and Visual Transformer Attention (VTA) as shown in fig.20. The first component is the Backbone Network. The backbone network is a standard image classification network, without the final classification layer. Some auxiliary layers are added to this backbone network to produce detection predictions. This component can employ deep learning models, such GoogleNet, ResNet, or EfficientNet as a backbone network. The second component is Grid-Wise Attention. GWA is used to mitigate the weakness of convolutional filters. The third component is Visual Transformer Attention. It converts the features acquired from advanced convolutional filters into a sequential arrangement of visual tokens. The Token-based Visual Transformer then processes this token sequence to acquire a comprehensive representation for FER tasks.

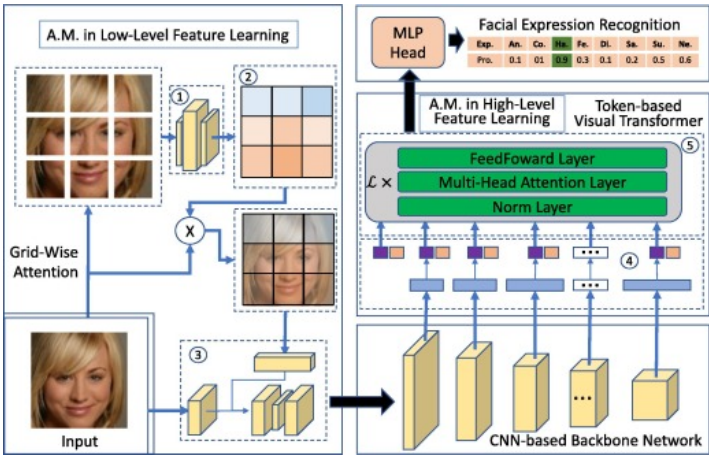

A.M. in Low-Level Feature Learning
1
2
Grid-Wise Attention
X
3
Input
MLP Head
Facial Expression Recognition
Exp. An. Co. Ha. Fe. Di. Sa. Su. Ne.
Pro. 0.1 01 0.9 0.3 0.1 0.2 0.5 0.6
A.M. in High-Level Feature Learning
Token-based Visual Transformer
5
$\mathcal{L}$ ×
FeedFoward Layer
Multi-Head Attention Layer
Norm Layer
4
CNN-based Backbone Network

fig.20

Paper Link: Facial expression recognition with grid-wise attention and visual transformer.

Datasets: The model was tested using datasets that include: CK+, FER+, and RAF-DB.

Model Source Code: https://github.com/ZBigFish/FER-VT

Model Weights: We found model weights in the github repo.

Information about the frameworks used in the code:

The model uses Python 3.6+,pytorch 1.0.0+ as mentioned in the github repo.

Information about training resources: Not found.

Results:

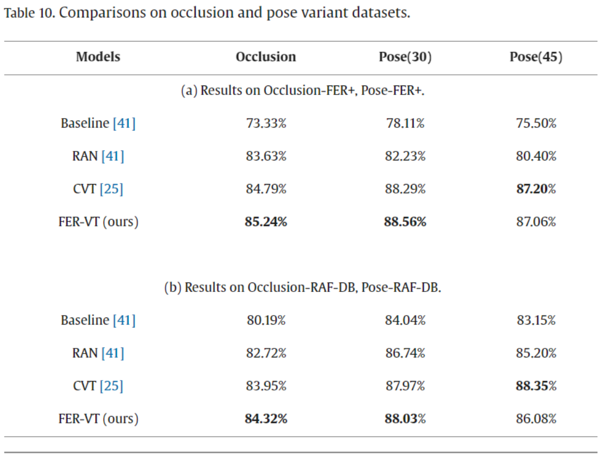


Table 10. Comparisons on occlusion and pose variant datasets.

Models	Occlusion	Pose(30)	Pose(45)
(a) Results on Occlusion-FER+, Pose-FER+.			
Baseline [41]	73.33%	78.11%	75.50%
RAN [41]	83.63%	82.23%	80.40%
CVT [25]	84.79%	88.29%	**87.20%**
FER-VT (ours)	**85.24%**	**88.56%**	87.06%
(b) Results on Occlusion-RAF-DB, Pose-RAF-DB.			
Baseline [41]	80.19%	84.04%	83.15%
RAN [41]	82.72%	86.74%	85.20%
CVT [25]	83.95%	87.97%	**88.35%**
FER-VT (ours)	**84.32%**	**88.03%**	86.08%

fig.21

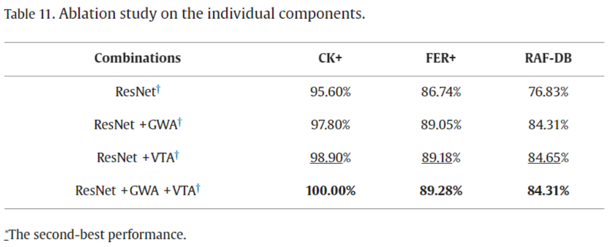


Table 11. Ablation study on the individual components.

Combinations	CK+	FER+	RAF-DB
ResNet†	95.60%	86.74%	76.83%
ResNet +GWA†	97.80%	89.05%	84.31%
ResNet +VTA†	98.90%	89.18%	84.65%
ResNet +GWA +VTA†	**100.00%**	**89.28%**	**84.31%**

_The second-best performance.

fig.22

From the results shown in fig 21 and 22, the model is achieving amazing results compared to other models

Third, Attentional Convolutional Networks.

Introduction to the model:

The Attentional Convolutional Networks model was introduced in the paper: Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network .Researchers noticed that traditional approaches to the facial expression recognition like SIFT, HOG and LBP perform well on images captured in a controlled condition ; however, they do not perform as good in more challenging images. They also noticed that despite the fact that CNNs (Convolutional Neural Networks) are capable of learning many features from decent images. However, traditional CNNs do not take into account that much of the clues come from specific parts of the images like the mouth, eyes, etc while other parts like hair play a little role in output prediction. Researchers built on the model based on the fact that some parts of the face contribute to facial expressions more than others. As a result, they proposed a deep learning based framework that takes the above observation into consideration. In order to do so, they used a CNN in addition to an attention mechanism that focuses on the salient parts of the face. They showed in this paper that using their proposed model even with less CNN layers, the model was capable of achieving high accuracy rates.

Architecture:

The researchers proposed a deep learning model based on attentional neural networks to classify the underlying emotion in the faces images. To make use of the fact that some parts of the images contribute more to the prediction of the expression, researchers added an attention mechanism through a spatial transformer network. In the model architecture, the feature extraction part consists of 4 convolutional layers, each two of the convolutional layers are followed by a max pooling layer and a Relu layer. Then , they are followed by two fully connected layers. In parallel to the four convolution layers, the spatial transformer or the localization network, consists of two convolutional layers and two fully connected layers. After the two fully connected layers, the transformation parameters are regressed and the input is transformed into a sampling grid T(**θ**) which produces the wrapped data. The role of the spatial transformer is to mainly focus on the relevant parts of the image, be sample estimation over the attended region. Then, an affine transformation was used to wrap the input to the output. The model is then trained by applying optimization to the loss function using a stochastic gradient descent approach. The loss function in this model is the classification loss function, cross entropy, + the regularization term, L2 regularizer.

$$\mathcal{L}_{overall} = \mathcal{L}_{classifier} + \lambda \| w_{(fc)} \|_2^2$$

It is worth mentioning that researchers added up to 50 layers but the accuracy did not improve much. As a result, they ended up with the architecture depicted in the figure below.

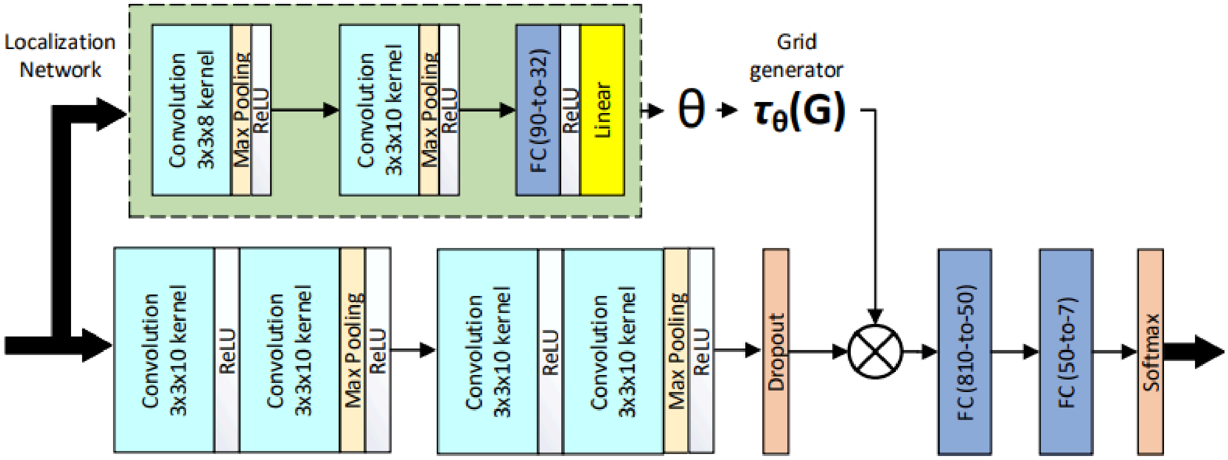


Fig. 3: The proposed model architecture

fig.23

Paper Link: [Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network]().

Datasets: Researchers trained and tested their model on the following datasets: FER2013, CK+, Japanese female facial expressions (JAFFE), and the facial expressions Research Group database (FERG).

Model Source Code: https://github.com/omarsayed7/Deep-Emotion

Model Weights: No mention of the model weights in the github repo.

Information about the frameworks used in the code:

The code was mainly written in the following library:

1. Pytorch >= 1.1.0
2. Torchvision == 0.5.0
3. Opencv
4. tqdm
5. PIL

Information about training resources: Not found.

Results:

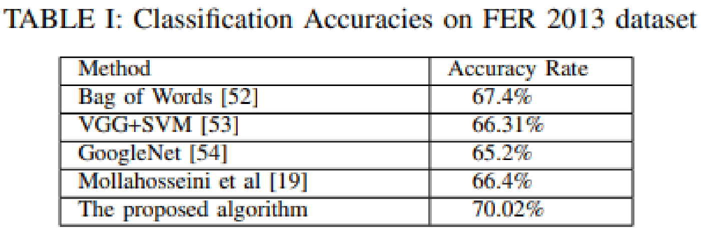


TABLE I: Classification Accuracies on FER 2013 dataset

Method	Accuracy Rate
Bag of Words [52]	67.4%
VGG+SVM [53]	66.31%
GoogleNet [54]	65.2%
Mollahosseini et al [19]	66.4%
The proposed algorithm	70.02%

fig.24

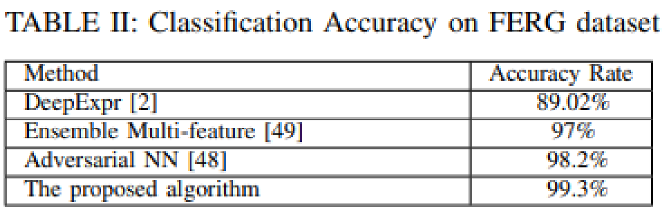


TABLE II: Classification Accuracy on FERG dataset

Method	Accuracy Rate
DeepExpr [2]	89.02%
Ensemble Multi-feature [49]	97%
Adversarial NN [48]	98.2%
The proposed algorithm	99.3%

fig.25

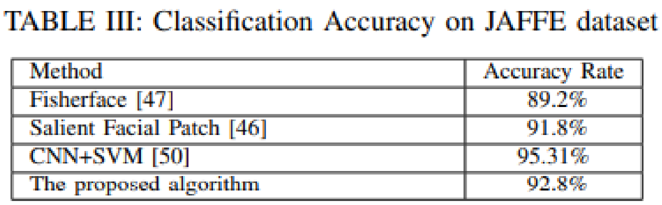


TABLE III: Classification Accuracy on JAFFE dataset

Method	Accuracy Rate
Fisherface [47]	89.2%
Salient Facial Patch [46]	91.8%
CNN+SVM [50]	95.31%
The proposed algorithm	92.8%

fig.26

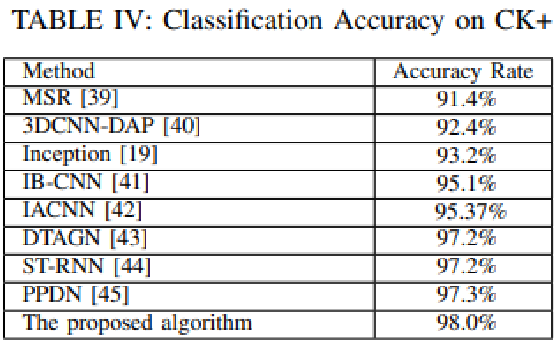


TABLE IV: Classification Accuracy on CK+

Method	Accuracy Rate
MSR [39]	91.4%
3DCNN-DAP [40]	92.4%
Inception [19]	93.2%
IB-CNN [41]	95.1%
IACNN [42]	95.37%
DTAGN [43]	97.2%
ST-RNN [44]	97.2%
PPDN [45]	97.3%
The proposed algorithm	98.0%

fig.27

From the above results, we can see that this model is doing well relative to some state of the art models.

Comparison between the above models:

It can be seen from the results of the three models that there was no common dataset between all of them. However, in the datasets that are common between two of them, the performance of both models is extremely close.

Selected Baseline Model.

From our literature review, we surveyed three models, explained their architectures, discussed the problem they addressed and reported their results. After careful consideration of all the three models, we choose the attentional CNN model for the following reasons:

1. It is the closest model to the direction that we are interested to discover which is adding an attention mechanism to a neural network to make use of the semantic importance of some parts of the face like eyes, mouth, eyebrow, etc.
2. We are more confident about our understanding of the architecture of this model than the other model as it is simpler and more straightforward. As a result, we were more able to come up with proposed updates for this model than the other models.
3. When we went through the implementation of the three models, we felt more comfortable when discovering the code of the attentional CNN model and we were able to understand and identify the flow of the code of this model. Other models were somehow more complicated and we faced difficulties when trying to understand them.
4. This model is much simpler in terms of architecture and the depth of the layers than the other two models. As a result, it will take less time in training than the other models.
5. This model has a built-in support for the FER2013 dataset which is one of the two datasets that we decided to use.
6. The selected model satisfies all the requirements as it has less than 10 files and is written in recent frameworks.

Detailed Description of the selected model: We have already described the selected model in full details in the previous section along with its baseline code and all the important information about the model.

How will we approach the model without the weights:

To address the issue of having no trained weights of this model, we have put a list of solutions:

- We have managed to get the contact information of the author of the github repo containing the baseline code. We will attempt to contact him very soon asking him if he has a pre-trained version of the model or the model weights.
- We intend to use the computers in the Machine Learning Lab because of its high computational power, especially the GPUs, relative to our computers. We will try to train the baseline model as soon as possible to get the trained weights.
- We intend to purchase computational units in Google CoLab to make use of its GPUs and memory.
- Not having the weights is not a hassle for us as the architecture of the model is simple and not hugely deep. Also, the datasets we chose are not big at all in terms of size. As a result, we believe that not having the weights will not hinder from producing a good quality project.

Proposed Updates to the model:

After doing the models survey and forming a good understanding of their natures and architectures, we propose the following updates to our baseline model:

First, Trying to use the Grid Wise attention (GWA) mechanism proposed in the FER-VT model instead of the spatial transformer.

As mentioned in the paper of the FER-VT model, adding the GWA mechanism was vital for the success of this model. As a result, we intend to replace the spatial transformer with the GWA mechanism to see what effect it can have on the results of our model. Also, we will search for other attention mechanisms to use in our model.

Second, Implementing a mechanism to handle the issue of noisy labels in the training data.

In the first paper in our survey, it was proved that having noisy labels can have devastating effects on the model performance. As a result, we will try to develop a mechanism of handling the noisy labels and limit their contribution in the training.

Third, Fine tuning of the model hyperparameters.

Since there was no clear mention in the paper of our baseline model whether they fine tuned the model hyperparameters or not, we will try to fine tune the parameters to get better results. Fine tuning will include the activation functions, using another loss function other than the cross entropy, other regularization methods, etc.

Fourth, Usage of Adaboost ensemble classifiers.

As ensemble classifiers have a potential positive impact on the performance of the model, we intend to use an ensemble classifier, specifically Adaboost, to see whether it can positively impact our model or not. The classifiers will be various versions of the baseline model. Differences between them will be related to the attention mechanism and model hyperparameters.

Finally, Data Augmentation.

Since the chosen dataset has imbalanced data as some classes clearly outnumber the other classes, we will see the possibility of using data augmentation to get a better performance of the model.

Comparison between our project and the previous years projects.

After going through the previous years projects, we discovered that some projects were also about facial expressions recognition. However, our project and direction clearly differs from those projects in the previous years in the following points:

- We are exploring the effect of adding an attention mechanism to the CNNs on improving model performance. Other projects' focus was mainly about the architecture of the CNN itself without any attention layers.

- The baseline models of the previous projects were VGG 16, Inception V3, Res Net, FeatX, Dexpression models, DCNNs. We used a completely different baseline model which is Attentional CNN.
- We proposed novel updates like implementing a mechanism to handle the noisy labels, usage of GWA (Grid Wise Attention) Mechanism, and using Adaboost ensemble classifiers. Those updates were not mentioned in any previous projects.

How to evaluate our results.

Earlier in the evaluation metrics section, we discussed a lot of metrics that are useful in our problem. From these metrics we will depend mainly on the following:

1. Accuracy: It is the main and the most important evaluation metric for our problem as it is the metric used to decide on the state of the art performances in the Facial Expression Recognition Problem. Also, it is used in every model and in every paper that attempts to solve the FER problem.
2. Confusion matrix: Confusion matrix is of extreme importance in our project as it will enable us to check the model's performance on the different classes. So, it will enable us to see whether the model is not doing well in a specific class or not. In other words, the confusion matrix acts as a breakdown for the accuracy. Finally, Many of the papers that we researched used the confusion matrix as an evaluation result for their work
3. Precision and recall: Precision and recall are also essential for our evaluation as they will enable us to check the model's performance in each class.
4. ROC Curve: We will try to plot the ROC curve and get the area under it for more evaluation for our model.

Briefly, the most important metrics that we will focus on the most are the accuracy and the confusion matrix. The other metrics (precision, recall and accuracy) wil be used for more evaluation for our model but they will not be the basic metrics.

Graduation Project Statement.

None of us is graduating this semester.

Previous Data science Projects Done:

Youssif: I have done a project in the Fundamentals of machine learning course. The project was about building a neural network for predicting the used cars prices in the US. This project is far away from the scope of this project. Although, I have used a neural network in the previous project, it was not an attentional CNN and it was a regression problem not classification like this one.

Sara: I have taken a statistical machine learning class during my semester abroad at Brandeis University and for the class project I worked on a Novozymes Enzyme Stability Prediction model (kaggle competition). I have worked on a Copper Future price prediction deep learning model using LSTMs as a part of my internship at Qara Digital Solutions. I have also worked on another model that predicts hit songs using repeated chorus.

Each Member Contribution:

Youssif: Did the survey about the available evaluation metrics, and the current state of the art models. Did the survey on the EAC model and the Attentional CNN model.

Sara: Did the datasets survey. Did the survey on the FER-VT model.

Both of us: Decided on the datasets to use, the baseline models. We also thought about the updates together, and wrote the introduction and the motivation part.

References

Accuracy. Hasty.ai Documentation. (n.d.). https://hasty.ai/docs/mp-wiki/metrics/accuracy#:~:text=Accuracy%20score%20formula,-Fortunately%2C%20Accuracy%20is&text=The%20Accuracy%20score%20is%20calculated,by%20the%20total%20prediction%20number.

AffectNet. Mohammad H. Mahoor, PhD. (n.d.). http://mohammadmahoor.com/affectnet/

Challenges in representation learning: Facial expression recognition challenge. Kaggle. (n.d.). https://www.kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge/data

Duta, S. (2023, January 30). *CK+-Dataset*. Kaggle. https://www.kaggle.com/datasets/stefanaduta/ckdataset

Huang, Q., Huang a b, C., Wang a, X., & Jiang a, F. (2021, November). Facial expression recognition with grid-wise attention and visual transformer. https://www.sciencedirect.com/science/article/abs/pii/S0020025521008495?fr=RR-2&ref=pdf_download&rr=80cdd0046b7441fc

Minaee, S., & Abdolrashidi, A. (2019, February 4). *Papers with code - deep-emotion: Facial expression recognition using attentional convolutional network*. Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network | Papers With Code. https://paperswithcode.com/paper/deep-emotion-facial-expression-recognition

omarsayed7. (n.d.). *Omarsayed7/deep-emotion: Facial expression recognition using attentional convolutional network, pytorch implementation*. GitHub. https://github.com/omarsayed7/Deep-Emotion

Saini, A. (2023, September 20). *ADABOOST algorithm: Understand, implement and master AdaBoost*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/

Segal, N. (2023, January 16). *Facial expressions training data*. Kaggle. https://www.kaggle.com/datasets/noamsegal/affectnet-training-data?select=labels.csv

ZBigFish. (n.d.). *ZBigFish/Fer-VT: The unofficial implementation of paper "facial expression recognition with grid-wise attention and visual transformer."* GitHub. https://github.com/ZBigFish/FER-VT

Zhang, Y., Wang, C., Ling, X., & Den, W. (2022, July 21). *Papers with code - learn from all: Erasing attention consistency for noisy label facial expression recognition*. Learn From All: Erasing Attention Consistency for Noisy Label Facial Expression Recognition | Papers With Code. https://paperswithcode.com/paper/learn-from-all-erasing-attention-consistency

Zyh-Uaiaaaa. (n.d.). *Zyh-UAIAAAA/erasing-attention-consistency: Official implementation of the ECCV2022 paper: Learn from all: Erasing attention consistency for noisy label facial expression recognition*. GitHub. https://github.com/zyh-uaiaaaa/erasing-attention-consistency