Emotion Detection from facial images on three different models

M Jamil Zaitouny
zjie2644@scs.ubbcluj.ro

Mihail Dorin Iliescu
imig0243@scs.ubbcluj.ro

Andreea Coaja
caig0228@scs.ubbcluj.ro

February 3, 2021

Abstract

Emotions are of key importance when it comes to collective understanding among people. In recent years, emotion recognition has become a vital component in the field of affective computing. Among many physiological and kinematic signals that could be used to recognize emotions, acquiring facial expression images is one of the most natural and inexpensive approaches. [22] The challenges that must be taken into consideration when talking about emotions are anatomical, cultural, and environmental differences. It is common knowledge that facial expression for emotion detection made by humans is an easy task, but what about achieving the same task with a computer algorithm that does not have empathy? Having this question in mind, we want to make use of a inception-based technique that has been developed during the last years for getting a better result regarding emotions detection. In this paper, we will use facial emotion recognition using convolutional neural networks Inception V3 [25], MobileNetV2 [23] and ResNetV2 with KDEF [7],AffectNet [18] and FER-2013 [2] as datasets.

1 Introduction

Facial detection is a subsegment of object detection which is a part of the field of computer vision. While the aim of object detection is to detect many objects, facial detection is more specific because it is only focused on peoples' faces. Face detection can be defined through the following scenario: Given a photo, the purpose of face detection is to determine if there are any faces present in the image, and if so, return the region containing the face.

There has been great interest in developing facial recognition systems for detecting facial emotions, since society has been tending to use social media more often rather than meeting with somebody face to face, which makes it harder for people in general to differentiate when a person has changed their emotional state. Therefore, the question of whether face detection could be used to detect different emotions with a much higher accuracy than humans is worth investigating. There are a few approaches that have been published that we are using as a point of reference.

1.1 Motivation

Nowadays people are spending substantial amounts of time on social media applications, 16-24-year-olds being the biggest culprits at around 3 hours a day [4], making it harder for them to distinguish emotions on other people's faces [24]. Technology is meant to bypass the limitations that human beings have, and a more specific problem regarding emotion detection is aggregating our aim is to try to take human evolution into own hands by investigating the accuracy of different convolutional models when it comes to detecting people's feelings from front images.

Our scope is to see what differences can be for detecting emotions from images while using three different pretrained models: MobileNetV2, InceptionV3 and ResNetV2. We want to see the differences in the results from each model. The labels for emotions used in this research are Anger, Disgust, Fear, Happiness, Sadness, Surprise and Neutral.

1.2 Context

Given the actual context of the corona crisis, we thought that it would be the perfect opportunity for us to track the emotion states of students throughout a lecture because most education institutes have implemented e-learning. It is something of interest at this moment because schools and universities want to improve the learning systems so that students can retain as much information as possible. We are hoping that our work will be able to provide teachers insight into the states of the students in hopes of providing the teacher with a good metric to evaluate their teaching style.

2 Key terminology

Ensemble: An ensemble uses multiple deep learning models to obtain a better predictive performance, usually by weighing the quality of the predictions of different models and averaging them depending on the weight.

Classification: Classification is one of the tasks of computer vision which uses convolutional neural network models to sort images into different preset classes.

Data pre-processing: Involves changing the dataset in some way to improve knowledge discovery, in our case, we extracted the faces from the images, we balanced the classes, we removed bad extractions, we also reduced the size and removed unnecessary features from the images by turning them into gray-scale.

Object extraction: Extracts wanted objects from an image by using localization techniques.

Arousal and Valence: A dimensional model used to detect the emotional granularity on a specific face. Arousal being the intensity of a specific emotion and valence being the quality of an emotion ranging from positive to negative.

Underfitting: Is when a model is not complex enough to accurately match all the features of a sample with the corresponding class during training.

Overfitting: Is when a model reaches a high level of accuracy on the training set at the expense of decreasing the ability of the model to accurately classify new samples.

3 Related work

In 2019, a study [19] investigating the classification of 7 different emotions used LeNet CNN architecture [12] and a mixture of datasets (KDEF [15], JAFEE [16] and a custom dataset). The aim of their study was to obtain a better performance for emotion recognition through facial expression. For making this possible, they used some techniques of image processing before training the model. They used Haar Cascade to firstly detect the faces, followed by cropping the images and converting them to grey scale. With this approach the authors have achieved accuracy of 96.43% and validation accuracy of 91.81%. What is also to mention is that the sad emotion state has obtained the lowest accuracy among the other emotions. A weakness of this study is the fact that they used LeNet which is a fairly outdated model that goes against the general trend of going deeper in the field of convolutional neural networks. However they did test their model on multiple datasets which probably led their model to be more effective when working with datasets with slightly different distributions.

In the same year, a paper was written [3] for recognizing the same 7 emotions that were used in the previous study, but using different methods. Their approach is based on using 5 datasets (CK and CK+ [13], FER-2013 [8], The MUG Facial Expression Database [4], KDEF and AKDEF [7] and KinFaceW-I and II [24]) and a Convolutional Neural Network with 15 layers. The workflow of their analysis is the following one: face detection (Cascade Classifier), face cropping, preprocessing & data augmentation(using ImageDataGenerator offered by Keras [17]) and after that the resulted dataset is fed into CNN to predict the class. The results from this work are very similar to the first one, with an accuracy of 96.24%. The model created in this paper might potentially work good in more general workloads when compared to the LeNet paper discussed previously, however a major weakness of this paper is that they were testing on posed images, which are really easy to overfit to.

One year later, in 2020, was published a paper [9] which has as main scope a comparison between datasets for prediction of emotions and valence arousal. For this, they have used 3 datasets: AffectNet [18], Aff-Wild [27] and AFEW-VA [11] and a proposed network based on YOLOv2 architecture with 17 layers. What is of interest for us is the accuracy for emotion detection which was made on AffectNet. Their approach achieved an accuracy of 75%.

Ensembles have been proven to significantly improve accuracy and sensitivity in some deep learning workloads [14]. There has been some attempts of using ensembles for emotion recognition, however in a lot of the cases most of the models that were used were custom made and didn't contain state of the art deep learning features such as inception blocks or skip connections from ResNetV2s [1]. There are however papers [6] that successfully used ensembles for image classification, and we're planning on building on those to advance the field of affective computing.

4 Pre-Trained Neural Networks Used in the Paper

Our scope is to train three models in order for us to make a conclusion about which one is the most appropriate for this task of emotion recognition.

4.1 MobileNetV2 model

MobileNetV2 [23] is an improved version of MobileNet [10] that significantly improves its accuracy. The MobileNetV2 network architecture consists of 19 blocks named bottleneck residual blocks (see Figure 1). MobileNetV2 uses depth-wise separable convolutions as efficient building blocks. There are two types of blocks in the architecture: one is a residual block with a stride of one, and the other one is a block with a stride of two for downsizing (see Figure 1). There are 3 layers for both types of blocks as following: a 1×1 convolution with ReLU6, a depthwise convolution and a 1×1 convolution but without any non-linearity. Moreover, it has linear bottlenecks between the layers where the use of linear layers is essential, as it blocks nonlinearities from damaging too much information. These bottlenecks help the model in encoding the intermediate inputs and outputs. The inner layer helps in transforming lower-level concepts such as pixels to higher-level descriptors such as image categories. There are also shortcut connections between the bottlenecks.

A pre-trained MobileNetV2 was used in this paper (Figure A5). Transfer learning was used to optimize the model for emotion detection. After adding the layers from MobileNetV2, we have added an Avarage Pooling layer, Flatten layer, 3 Fully Connected layers followed each of one by one Dropout layer and a Batch Normalization layer (see Figure 2). We changed the classification layer so that the number of classes the network classifies became 7 (see Figure 2).

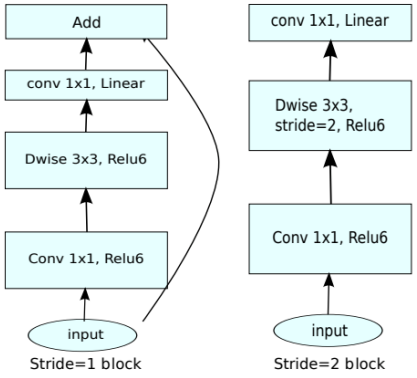

Figure 1: MobileNetV2 blocks

4.2 ResNetV2 model

[1] One of the reasons that ResNets are incredibly powerful due to their ability to go much deeper than other networks without facing problems such as diminishing/exploding gradient. This is done by utilizing skip connections

ResNetV2 is a improved variation of ResNet that applies Batch Normalization and ReLU activation to the input before the multiplication with the weight matrix in the skip connections. ResnetV2 also removes the last non-linearity, therefore, clearing the path of the input to output in the form of identity connection.

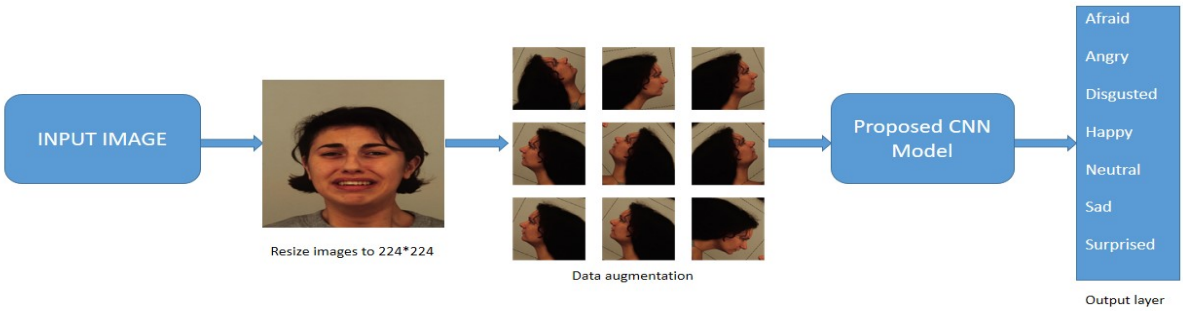

Figure 2: The flow for training data for emotion recognition

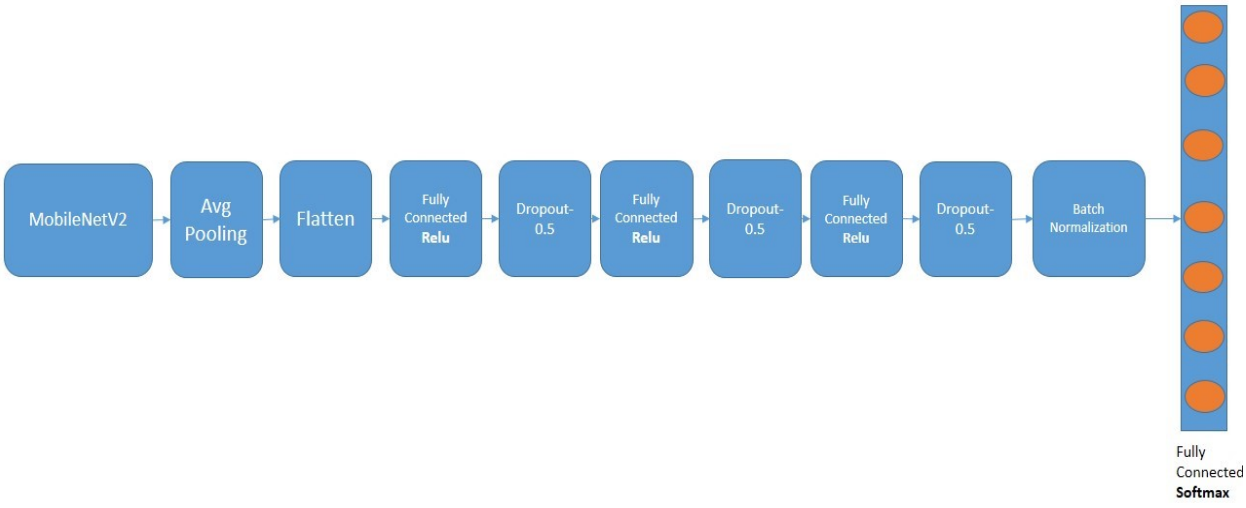

Figure 3: Proposed CNN Model used with MobileNetV2

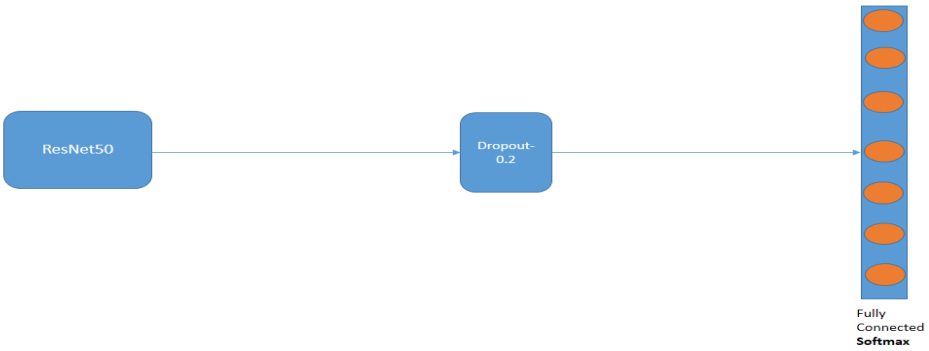

Figure 4: Proposed CNN Model used with ResNetV2-50

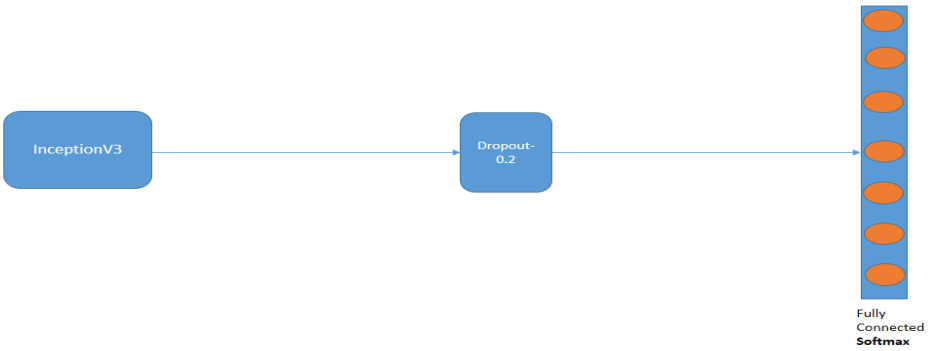

Figure 5: Proposed CNN Model used with InceptionV3

4.3 InceptionV3 model

It took inspiration from the approach based on primate visual cortex dictated by Serre et al. [20] which can handle multiple scales. One of the important criteria of Inception architecture is their adaption of "Network in Network" approach by Lin et al [26] which increased the representational power of the neural networks. This had additionally saved them for computational bottlenecks by dimension reduction to 1×1 convolutions. The purpose of Inception architecture was to reduce computational resource usage in highly accurate image classification using deep learning [21]. For such a problematic dataset we used SGD,which is a variant of gradient descent. Instead of performing computations on the whole dataset — which is redundant and inefficient — SGD only computes on a small subset or random selection of data examples. SGD produces the same performance as regular gradient descent when the learning rate is low.

5 Data preparation

5.1 Dataset Choice

Among the many prominent facial expression datasets we decided to use KDEF. After experimenting on the dataset with different models, we noticed that due to the low number of images in the dataset and the posed nature of the images; our model was overfitting heavily on KDEF. which led to our decision to also use the manually labeled image set from Affectnet to finetune our models.

AffectNet is a very large dataset consisting a huge number of different size colored images. Being such large scale dataset it is not free from faulty data, and we have seen some example in figure 3.3. But another important issue that arises is ambiguity among data. Some pictures in different classes look very similar, and they are bound to create confusion in the classification. For testing we decided to FER-2013. It's is a a dataset designed by Goodfellow et al. for a Kaggle competition to promote researchers to develop better facial expression recognition systems [5].

5.2 Ambiguity in the Dataset

The example below has 2 images from Angry and 2 images from Disgusted. Though one might label all of them as Disgusted due to their ambiguity.Such limitation imposed by the dataset has resulted into a lesser accuracy.

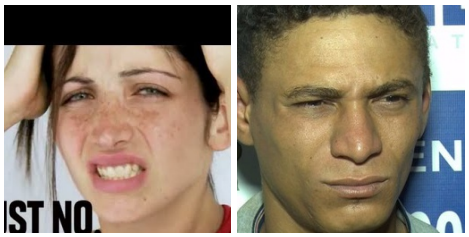

Figure 6: Files from folder angry but it can also be disgusted

AffectNet dataset is so frequent that human accuracy on labeling the images is very low as well. Ian Goodfellow has found out that the human accuracy on AffectNet dataset was 65±5

5.3 Dataset organization

We looked at the different approaches that were used and we found that most people train their models one dataset at a time and that there weren't many papers that combined datasets to improve the results of the classifiers. We decided that there needs to be more papers that have data mismatch, and that it would be a good idea to see why it isn't frequently used in computer vision papers.

For the data split, we decided to use the KDEF classes on Affectnet, as we were looking to test the accuracy of our model on KDEF. Which led us to extract 7 classes from Affectnet which were fear, anger, disgust, happy, neutral, sad, surprised and not using the other 4 classes that comes with Affectnet.

Due to computational power limitations we weren't able to train on all of Affectnet, so we ended up deciding to retrieved 4000 images per class to reduce training time. Although due to errors while applying HAAR cascading, we ended up with around 3200 images +-50 per class. We also extracted 100 images per class from KDEF to be added to our training set alongside of Affectnet, so the total of images for training was of 23100 images. For the

validation set we used the remaining images fom KDEF but only with the ones that have front pictures, which led to having a 160 image per class for validation, so a total of 1120 of pictures for validation dataset. For testing dataset we have used half of the images from Fer-2013 which means 1675 per class, so a total of 11725 images. There was a slight imbalance in the number of imager per class in KDEF with some classes having 159 images but we deem it to be insignificant to take any action on it. There was a more significant mismatch between the number of images in the Affectnet dataset, due to the disgust class having 199 images less than the other classes, to combat the data mismatch we decide to take add some of the images from the automatically labeled dataset to the one we're using to combat that mismatch. There were also an imbalance into the testing dataset with a number of +/-50 photos per class.

5.4 Data Augmentation

For data augmentation we rotated the images randomly between the rotation range of 45 degrees, we also randomly flipped the images on the horizontal axis. We also shifted some of the images by 20% of the width and/or height. The images were augmented on every epoch, using the ImageDataGenerator class, we also normalized the augmented images before every epoch by setting the featurewise_center and the featurewise_std_normalization options to true which helped speed up training.

6 Training the models

As described above, the paper follows to use transfer learning on three different models by using the same approach.

When it's about training the pretrained model we have come up with some techniques that can improve the accuracy of the models. In this way we have used data preprocessing, checkpoints and we have fine-tunned our models at the final.

6.1 Callbacks

In this sense we have used three different callbacks to obtain the best accuracy that we can. We have added them progressively depending on the results that we have obtained after each training. Having this in mind, we have sticked to using Early Stopping, Model Checkpoint and ReduceLROnPlateau. The training for finding the best combination of results was made on the MobileNetV2 model and by using KDEF and AffectNet as datasets (around 3400 images per class for the training and for the validation set we used around 150 images per class, for the testing we used FER-2013 containg around 1500 images per class for the testing set).

As a first attempt we have added the Model Checkpoint and ReduceLROnPlateau. Model Checkpoint is used to save the weights of the model with the best validation loss value. ReduceLROnPlateau was used to automatically change the value of the hyperparameter rate from Adam's optimizer. As a start we have used Adam with a rate of 0.02 and ReduceLROnPlateau with a factor of 0.1 and a minimum of 0.0001 and just half of the training_datset. The results were not the best as we got only 38% on the test_dataset. After that we have changed the value of the rate from Adam's optimizer with 1e-4, trained the model with the whole dataset and adding early stopping checkpoint so that the training will stop when the validation dataset starts to degrade.So, in this case, we got a much better result, of 40.87% after the Early Stopping has stopped the training after the 54th epoch.

6.2 Fine-tuning

In order to make use of transfer learning, we have freezed the whole base model at first. After training the mobilenet we have unfreezed the last 56 layers from the base model so that we could fine tune our results. Moreover, we have chosen a number of 25 epochs, the value for the learning rate was 0.00001 (lower than the minimum value from ReduceLROnPlateau) and we used just the Model Checkpoint so that we could save the best values for the weights of the models. After this part, we got an accuracy of 76.7% on the training dataset.

However we noticed that freezing parts of the network/the whole network didn't work as well on ResNetV2-50V2/InceptionV3, we ended up fine tuning on the whole model for the best accuracy.

6.3 First Training

Initially we trained the data using the following parameters:
• Initial Learning Rate = 1e-4 being modified each time a plateau was reached with the help of ReduceLROnPlateau checkpoint
• Epochs number = 100 but it was necessary only 75 epochs by applying Early Stopping checkpoint
• Batch Size = 32
• Step Size = 755
• Colored photos (AffectNet+100KDEF containing 23480 images in total (100 KDEF per class and the rest from Affectnet) which was used for training, KDEF-100 for validation and for testing we used the FER-2013 dataset)
• Adam Optimizer

Note: When we first started we used a learning rate of 0.01 and RMSProp as an optimizer, we learned through iteration that they weren't optimal for our model. We also realized that MobileNetV2 was heavily underfitting on the training set, So we decided to add more layers as shown in figure 3. We attempted to do the same thing for InceptionV3 and but we didn't see the same performance boost as we did with MobileNetV2, and the computational cost was too high to justify spending a lot of time waiting on those models to train.

6.3.1 Fine tuning

Updated parameters for fine-tuning:
• Learning Rate = 0.00001
• Epochs number = 25
Note: As mentioned in section 6.2 we ended up fine-tuning incpetionV3/ResnetV2 on the whole model.

6.3.2 Result and Analysis after First Training

After our training we have evaluated the test set images. It resulted in an accuracy of 75.00% on the proposed model which used MobileNetV2 architecture.

Confusion Matrix of Test Set Evaluation after the First Training In the confusion matrix, the numbers highlighted in bold is the recall of different classes. Recall is percentage of positive cases that were labeled correctly by the classifier.

According to Figure 5, confusion matrix, proposed MobileNetV2 model is more accurate at prediction of disgusted and surprised emotion states with a value of 100%, and less accurate at prediction of sad emotion state (33%). However, what is remarkable is that we obtained a better prediction for sad emotion state compared to the research paper that we disscused above [19] for the same emotion with a value of 15%, more than doubled. But when giving a closer attention to the results, we could say that the model is overfitting.

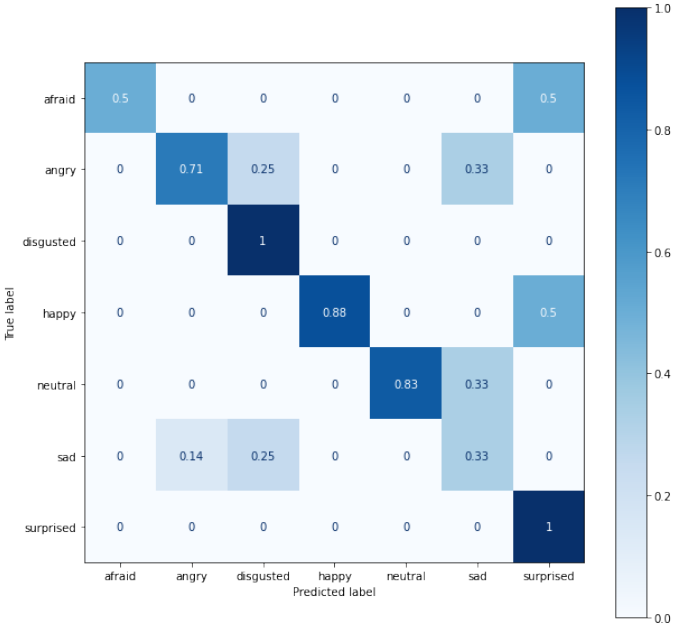

Figure 7: Results from MobileNetV2 1st Phase of Training

6.4 Second Training

In the 2nd part we greyscaled 1 channel in order to run faster and to have a better accuracy. The effect has been seen in 10% increase in validation accuracy , 10% decrease in loss function and up to 30 min quicker in the first epoch.

Moreover we changed Adam to SGD for a better accuracy.
• Initial Learning Rate = 0.005
• Epochs number = 100
• Batch Size = 32

• Step Size = 515
• Greyscaled photos (AffectNet+100KDEF, KDEF-100)
• SGD Optimizer

6.4.1 Result and Analysis after Second Training

After evaluating the test set images based on our second training we had an accuracy of 81.00

And we got the following confusion matrix:

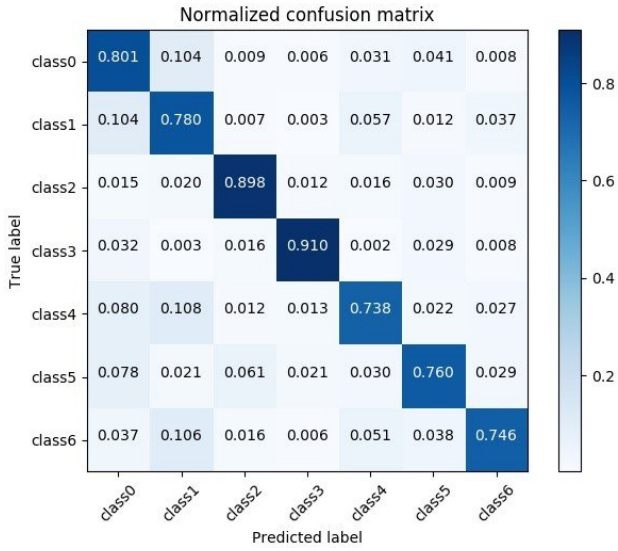

Figure 8: Results from InceptionV3 3rd phase testing

Confusion Matrix of Test Set Evaluation after the Second Training We can now see that the recalls are better than the inaccurate measurements, a issue we have previously faced on the first training.

6.5 Third Training

In the 3rd training we took advantage of the brand new and free Google Colab TPUs with the folowing configuration:
• Initial Learning Rate = 0.005
• Epochs number = 100
• Batch Size = 128
• Step Size = 72

Model	Accuracy	Loss
MobileNetV2	75%	1.26
ResNetV2	43%	1.12
Inception	80%	1.19

Table 1: Results on test set

• Greyscaled + Face cropped (Haarcascade) photos (AffectNet+100KDEF, KDEF-100)
• SGD Optimizer

6.5.1 Result and Analysis after Third Training

After evaluating the test set images based on our second training we had an accuracy of 87.00

And we got following confusion matrix:

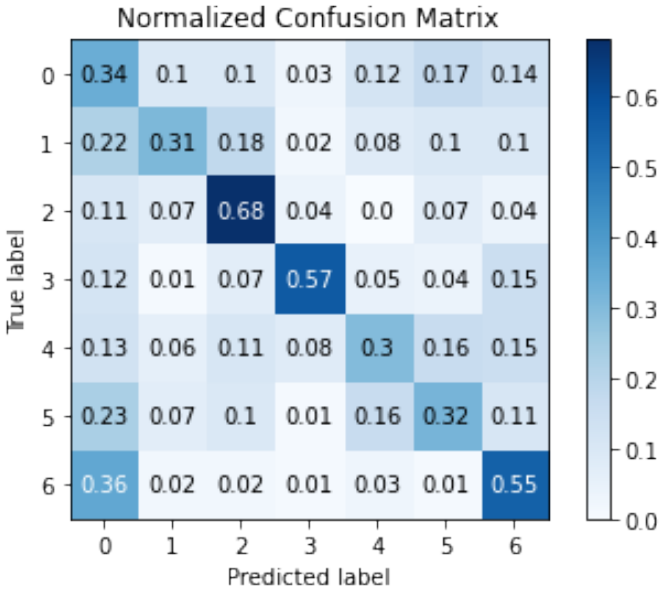

Figure 9: Results from ResNetV2 3rd Phase of Training

Confusion Matrix of Test Set Evaluation after the Second Training We can now see that the recalls are better than the inaccurate measurements, a issue we have previously faced on the first training.

6.6 Attempting to combine the models in one ensemble

Ensembling is a machine learning technique that works by combining predictions from two or more separate models. The most popular ensembling methods include boosting and bagging. Boosting works by using simple base models to increase their aggregate complexity. It trains a large number of weak learners arranged in a sequence, such that each learner in the sequence learns from the mistakes of the learner before it.

The other ensembling method is bagging, which is the opposite of boosting. Bagging works by training a large number of strong learners arranged in a parallel pattern and then combining them to optimize their predictions.

In our paper we focused on the latter ensembling method called bagging, we used a weighted average to combine predictions of the three models into one in hopes of achieving better generalized prediction and to reduce overfitting. On our first attempt we tried to use a common formula to rank the weights of the models.

$$w_i = \frac{R(A_i)}{\sum R(A_i)}$$

Figure 10: Formula used for the ranks

However we realized that some overfitting is being done in some of the models which reduced the quality of the ensemble, so we decided to manually adjust the weights for better results.

7 Conclusions

As a comparison to the related works we can say that our final results for the models are not better than the first two research papers, [19] with an accuracy of 96.43% and [3] with an accuracy of 96.24%. One reason for that is because we heavily relied on images in the wild for our training, whereas those papers relied mostly on posed images, we suspect that if we compared our models to their models on a new distribution our results will work better, as we managed to train on a larger, more general datase. But what is to mention, is that the accuracy over the sad emotion is better valued on our models as it can be seen in the Confusion Matrix of our trainings. In comparison to the last related paper, [9], which has also used the AffectNet as training data, we managed to obtain better result for ResNet50 and InceptionV3.

8 Contributions

- Optimizers - J.Z. and D.I.
- Loss - all of us
- Evaluation - all of us
- Checkpoints - A.C.
- Data preprocessing - J.Z. and D.I.
- Fine tuning - all of us
- Ensamble - J.Z. and D.I.
- Chapter 1 - A.C. and J.Z.
- Chapter 2 - A.C. and J.Z.
- Chapter 3.1 - A.C.
- Chapter 3.1 - J.Z.
- Chapter 3.1 - D.I.
- Chapter 4 - D.I. and J.Z.
- Chapter 5 - all of us
- Chapter 6 - all of us

References

[1] Detailed guide to understand and implement resnets https://cv-tricks.com/keras/understand-implement-resnets/. 3

[2] Fer-2013 https://www.kaggle.com/msambare/fer2013?select=train 1

[3] T. U. Ahmed, S. Hossain, M. S. Hossain, R. ul Islam, and K. Andersson. Facial expression recognition using convolutional neural network with data augmentation. pages 336–341, 2019. 2, 10

[4] N. Aifanti, C. Papachristou, and A. Delopoulos. The mug facial expression database. pages 1–4, 2010. 1, 2

[5] Amil Khanzada, Charles Bai, and Ferhat Turker Celepcikay. Facial expression recognition with deep learning. *CS231 Report*, 2020. 6

[6] AMIT KUMAR DAS, SAYANTANI GHOSH, SAMIRUDDIN THUNDER, ROHIT DUTTA, SACHIN AGARWAL, and AMLAN CHAKRABARTI. Automatic covid-19 detection from x-rayi mages using ensemble learning with convolutional neural network. *Choudhury school ofInfomration Technology Journal*, 2020. 3

[7] M. G. Calvo and D. Lundqvist. Facial expressions of emotion (kdef): Identification under different display-duration conditions. *Behavior research methods*, 40(1):109–115, 2008. 1, 2

[8] P. Giannopoulos, I. Perikos, and I. Hatzilygeroudis. Deep learning approaches for facial emotion recognition: A case study on fer-2013. pages 1–16, 2018. 2

[9] S. Handrich, L. Dinges, A. Al-Hamadi, P. Werner, and Z. Al Aghbari. Simultaneous prediction of valence/arousal and emotions on affectnet, aff-wild and afew-va. *Procedia Computer Science*, 170:634–641, 2020. 3, 10

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[11] J. Kossaifi, G. Tzimiropoulos, S. Todorovic, and M. Pantic. Afew-va database for valence and arousal estimation in-the-wild. *Image and Vision Computing*, 65:23–36, 2017. 3

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[13] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. pages 94–101, 2010. 2

[14] LUDMILA I. KUNCHEVA and CHRISTOPHER J. WHITAKER. Measures of diversity in classifier ensemblesand their relationship with the ensemble accuracy. *School of Informatics, University of Wales Journal*, 2003. 3

[15] D. Lundqvist, A. Flykt, and A. Öhman. The karolinska directed emotional faces (kdef). *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet*, 91(630):2–2, 1998. 2

[16] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with gabor wavelets. pages 200–205, 1998. 2

[17] K. Maksat, A. Lyazzat, and M. Mateus. Improved facial expression recognition with xception deep net and preprocessed images. pages 1–7, 2019. 2

[18] A. Mollahosseini, B. Hasani, and M. H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2017. 1, 3

[19] M. A. Ozdemir, B. Elagoz, A. Alaybeyoglu, R. Sadighzadeh, and A. Akan. Real time emotion recognition from facial expressions using cnn architecture. pages 1–4, 2019. 2, 8, 10

[20] Y. Pang, M. Sun, X. Jiang, and X. Li. Convolution in Convolution for Network in Network. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1587–1597, May 2018. 6

[21] Y. Pang, M. Sun, X. Jiang, and X. Li. Convolution in Convolution for Network in Network. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1587–1597, May 2018. 6

[22] M. Rescigno, M. Spezialetti, and S. Rossi. Personalized models for facial emotion recognition through transfer learning. *Multimedia Tools and Applications*, pages 1–18, 2020. 1

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. pages 4510–4520, 2018. 1, 3

[24] M. Shao, S. Xia, and Y. Fu. Genealogical face recognition based on ub kinface database. pages 60–65, 2011. 1, 2

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. pages 1–9, 2015. 1

[26] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. pages 1–9, 2015. 6

[27] S. Zafeiriou, D. Kollias, M. A. Nicolaou, A. Papaioannou, G. Zhao, and I. Kotsia. Aff-wild: Valence and arousal'in-the-wild'challenge. pages 34–41, 2017. 3