Real-Time Age, Gender, and Emotion Recognition Using Convolutional Neural Networks

Lian Duan (SID: 110058821) and **Dhruvil N. Shah (SID: 110089102)** and **Nimit D. Hingrajia (SID: 110091994)**
School of Computer Science
University Of Windsor

Abstract

Face recognition has become one of the most popular topics in computer vision and bio-metrics fields. Personal information, such as age and gender, are easily able to be abstracted and collected by using Deep Neural Networks. In this work, we utilize transfer learning and ST-CNN, which are trained by CK+, UTKFace, and FER2013 datasets, to classify age, gender, and emotion. With the help of image augmentation techniques and hyper-parameters tuning, ST-CNN achieves the best performance in emotion detection using the FER2013 dataset with an accuracy of 73%. Both ST-CNN and VGG-16 receive a remarkable performance in emotion detection using the CK+ dataset with an accuracy of 98%. For age and gender detection using the UTKFace dataset, VGG-19 achieves the highest accuracy in age classification with 66% and VGG-16 achieves the highest accuracy in gender classification with 91%.

Keywords: Face recognition, Transfer Learning, Spatial Transformer, Convolutional Neural Network, Image Classification, Computer Vision, Machine Learning

Acknowledgement

This project partially referenced the works of STN from GitHub https://github.com/kevinzakka/spatial-transformer-network.
Transfer Learning techniques, such as VGG16 and VGG19, were developed with the help of the work published by Simonyan, K., Zisserman, A. in "Very deep convolutional networks for large-scale image recognition" (2014). InceptionV3 and ResNet50 were implemented with help of research work published by Lin, M., Chen, Q., & Yan, S and He, K., Zhang, X., Ren, S., & Sun, J. in Network in network (2013) and Deep residual learning for image recognition (2016) respectively.

Introduction

Age, gender, and emotion detection are the techniques used in computer vision and artificial intelligence to automatically identify and analyze these features of individuals in digital images and videos. These techniques can be applied for a wide range of fields, such as security and surveillance, entertainment, marketing and advertising, and healthcare.

Facial expression detection involves identifying the emotions that a person is feeling based on their facial features. This can be done using machine learning algorithms that are trained on large datasets of facial images labeled with the corresponding emotions.

Gender detection, on the other hand, involves identifying the gender of a person based on their facial features. This can be done using similar machine learning algorithms, but trained on datasets of facial images labeled with the corresponding gender.

Age detection involves estimating the age of a person based on their facial features. This can be a challenging task, as the appearance of a person's face can change significantly over time. However, machine learning algorithms can still be used to achieve good accuracy in age estimation by training on large datasets of facial images labeled with the corresponding ages.

Overall, facial expression, gender, and age detection are powerful tools for automatically analyzing and understanding the facial characteristics of individuals in digital images and video.

Related Works

Singh and Nasoz (2020) described the recent implementation of Facial Expression Recognition using CNN. They demonstrate the process of classification by plugging static images into CNN architecture. They also provided some techniques to improve the accuracy of CNN prediction using pre-processing and feature extraction methods. After tuning and training the CNN model, they received a test accuracy of 61.7 % on the FER2013 dataset.

Minaee et al. (2021) proposed a new deep learning approach for facial expression recognition by using an Attentional Convolutional Network. Compared with other current deep learning approaches, such as CNN, they stated that the attentional approach is able to make a significant improvement in face recognition by focusing on important parts of the face. They also mentioned that they utilized visualization techniques to highlight the salient regions of face images, which are the most pivotal parts for distinguishing different facial expressions.

Xu et al. (2015) tested the model on occluded conditions as well and showed the ability of the model classification in small occlusion cases, Xu et al. (2015) also modified the model for improving the facial recognition on a self-build database and achieved 81.50 average accuracy.

Ng et al. (2015) stated that their experimental results show that cascading fie-tuning approach has achieved better results as compared to single-stage fine-tuning with the combined datasets with an overall accuracy of 48.5% and 55.6% in validation and testing dataset respectively.

Xue et al.(2021) propose the Trans-FER model which can learn rich relation-aware local representations. It mainly comprises three components: MultiAttention Dropping (MAD), ViT-FER, and Multi-head SelfAttention Dropping (MSAD). MAD is used to overcome the problem of overfitting the model. While ViT-FER is used for building relations between the different local patches.

To address the problem of multiple self attention that may extract similar relations the MSAD is proposed to randomly drop one self-attention module, increasing the diversity in relation.

FER in practical use poses unconstrained problems like detecting expression on partially occluded faces. In this paper, Li et al. (2018) propose a convolutional neural network (CNN) with attention mechanism (ACNN) that can perceive the occlusion regions of the face and focus on the most discriminative un-occluded regions. ACNN combines multiple regions from facial regions of interest (ROIs). Each representation is weighed via a proposed gate unit that computes an adaptive weight from the region itself according to the unobstructedness and importance. Considering different RoIs, Li et al. (2018) introduce two versions of ACNN: patch-based ACNN (pACNN) and global–local-based ACNN (gACNN). pACNN only pays attention to local facial patches. gACNN integrates local representations at patch-level with global representation at the image level.

Dataset

UTKFace

The UTKFace dataset is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over 20,000 face images with annotations of age, gender(male, female), and ethnicity(White, Black, Asian, Indian, and others). The proportion of each class in the UTKFace dataset is shown in Figure 1, Figure 2, and Figure 3

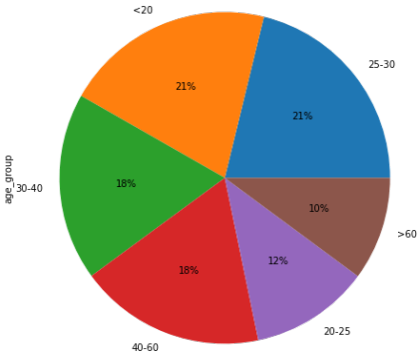

Figure 1: Age Group

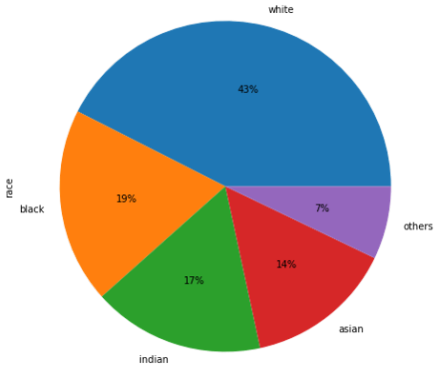

Figure 2: Race

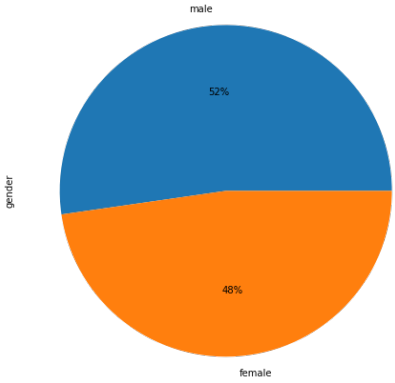

Figure 3: Gender

Extended Cohn-Kanade (CK+)

The Extended Cohn-Kanade (CK+) dataset originally contains 327 labeled with one of seven expression classes: anger, contempt, disgust, fear, happiness, sadness, and surprise. They are captured from 593 video sequences from a total of 123 different subjects, ranging from 18 to 50 years of age with a variety of genders and heritage. For this project, we utilize a modified version of CK+ which has eight classes: anger, contempt, disgust, fear, happiness, sadness, surprise, and neutral. The number of images of each class in the CK+ dataset is shown in Table 1.

FER-2013

FER2013 contains approximately 30,000 facial RGB images of different expressions with size restricted to 48×48, and the main labels of it can be divided into 7 types: anger, disgust, fear, happiness, sadness, surprise, and neutral. The number of images of each class in the FER2013 dataset is shown in Table 2.

Approach

We design experiments with different types of architectures that are well-known for image classification tasks. The high-level flow of our experiments is visualized in Figure 4. We divide our workflow into two groups: age and gender, emotion. Then, image data

Categories of emotions	Number of images
Anger	45
Contempt	18
Disgust	59
Fear	25
Happy	69
Sadness	28
Surprise	83
Neutral	593

Table 1: CK+ Dataset

Categories of emotions	Number of images
Anger	3995
Disgust	436
Fear	4097
Happy	7215
Sadness	4830
Surprise	3171
Neutral	4965

Table 2: FER2013 Dataset

is converted from RGB image to datafram in the preprocessing step, which can be processed by Python. Next, image augmentation takes unbalanced data as the input and randomly generates new images to fill the gap between each class. Finally, processed image data is fed into the models for training to generate classification outputs. Also, the models' best weights and configurations are recorded in order to recognize and classify the input image from the camera. The classification results, including age, gender, and emotion, are displayed on the screen. Beyond that, we also compare the results among the experiments and select those models which achieve the highest accuracy, F1 score, and lowest loss.

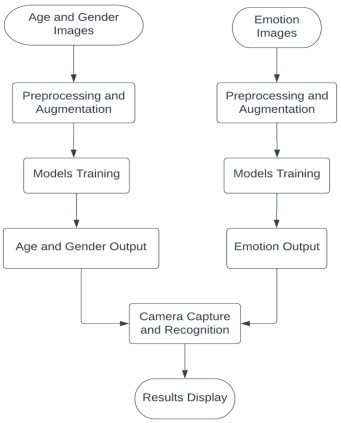

Figure 4: Proposed Approach

Computational Experiments

Experimental setup

There are two platforms we selected for our experiments: Jupyter Notebook with Python 3.9 and Kaggle Notebook with Python 3.7. As for the hardware, two dedicated GPUs are used to accelerate the training process, especially for Transfer Learning. Apart from that, we utilize four main toolkits and libraries: Tensorflow Keras, Scikit-learn, Nvidia CUDA, and OpenCV.

Evaluation metrics

Accuracy Accuracy is the ratio of correct predictions to total predictions. An accuracy metric is used to measure the algorithm's performance in an interpretable way and is usually determined after the model parameters and is calculated in the form of a percentage. It is the measure of how accurate the model's prediction is compared to the true data.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Loss A loss function is used to optimize a machine learning algorithm. The loss is calculated on training and validation, and its interpretation is based on how well the model is doing in these two sets. It is the sum of errors made for each example in training or validation sets. Loss value implies how poorly or well a model behaves after each iteration of optimization. As for the loss function, Cross entropy is used for evaluation in this project.

As the final part of our evaluation, the model was checked against the test set. The data required for this was created, and the model's 'evaluate' method in TensorFlow was used to compute the metric values for the trained model. This method returns the loss value and metrics values for the model in test mode.

F1 Score F1 score can be defined as the harmonic mean of the model's precision and recall. It is used to measure the rate of performance of a model.

$$F1Score = \frac{TP}{TP + 1/2(F + FN)} \tag{2}$$

Below are the abbreviations for the Formula

1. TP: True Positives
2. TN: True Negatives
3. FP: False Positives
4. FN: False Negative

Implementation Details

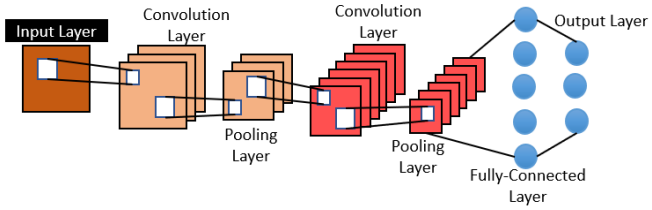

Figure 5: CNN

CNN CNN architecture is a deep neural network. It contains multiple hidden layers and each of these layers has several two-dimensional planes consisting of several neurons (He et al.,2016).

In addition to this, all neurons are assumed independently. In CNN input data can be considered a two dimensional image, and the feature extraction module is embedded with the CNN architecture. The basic architecture of CNN is shown in the Figure 5. The basic structure of CNN architecture can be divided into five main parts: input layer, Up-sampling convolution layer, Downsampling/pooling layer, fully connected and the output layer. The detailed explanation of each part is explained below.

Input layer: Raw data set can be directly inputted to the input layer. Image is inputted with its decided pixel value into the input layer.

Convolutional layer: It is also known as the upsampling layer which extracts the features from the inputted data. Each convolutional layer has its own convolutional kernels like (3x3 or 5x5) and these different convolutional kernels can extract different types of the features from the input data. This number of extracted features can grow as the number of convolutional kernels included in the up-sampling layer increases.

Down-sampling layer: It is also known as the pooling layer. This layer main function is to finish the second extraction of the feature data followed by the convolution layer. Basically, under the normal conditions, the CNN architecture contains at least two convolutional layers and two down-sampling layers respectively. With the more layers of the architecture are set, extracting features from input data are more likely to help obvious classification. (Guan et al.,2019)

Fully connected layer: All the available feature maps were connected and get transposed so that input the for a fully connected layer gets connected. Generally, this node of the neurons within the later layer is connected to the nodes of the neurons within the previous layer, but the nodes in every layer are disconnected. This layer integrates and normalizes the abstracted features of the previous convolutions in order to yield a probability for various conditions. (Guan et al.,2019)

Output layer: All the neurons in this layer are settled according to the required conditions. If the classification is required, the number of neurons is generally related to the number of categories to be classified. (Guan et al.,2019)

ST-CNN STN is a specialized type of CNN. It includes a number of spatial transformer modules that allow the model to recognize and classify features using data that is invariant to its input. Even when the input is marginally changed, invariance can aid the model in recognizing and identifying features. In other words, STN can increase overall performance while attempting to stabilize or clarify an object within a processed image or video. As a result, object classification and identification become more precise. Special transformer module can be introduced into convolutional networks that already exist. The spatial transformer has three primary components, as shown in Figure 6. The localization network is the initial component. To represent a set of photos as input, localization has width, height, channels, and batch size. Localization is a simple neural network consisting of few convolution layers and a few dense layers. Localization predicts the parameters of transformation as output, which is marked as Theta in Figure 6. These parameters determine several properties including rotation angle and scaling factor of the region of interest in the input images. Then, based on the parameters from the localization procedure, the grid generator constructs a grid of coordinates in the input image corresponding to each pixel from the target image. The transformation technique involved here is Affine transformation. Finally, the sampler employs bilinear interpolation to apply the transformation's parameters to the input image. In general, the input of a spatial transformer is a collection of images from datasets, and the output is a transformed feature map.

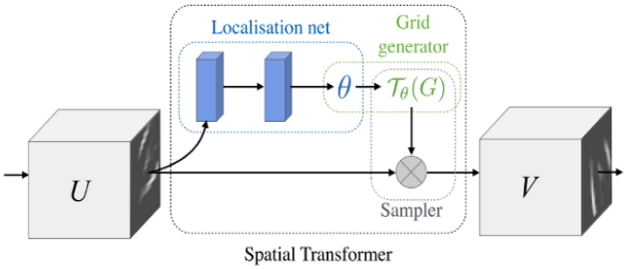

Figure 6: Spatial Transformer model (Jaderberg et al., 2016)

The diagram that illustrates how spatial transformer is incorporated in the CNN model is shown in Figure 7. To match the size of the affine transformation matrix, the picture size in the localization layer is reduced to 2x3. Both the input feature map and the sampling grid are processed in the grid generator and sampler layer, for example, using bilinear interpolation to generate a transformed feature map with a 48x48 size. CNN next begins to classify and categorize the photos into the number of classes we specified.

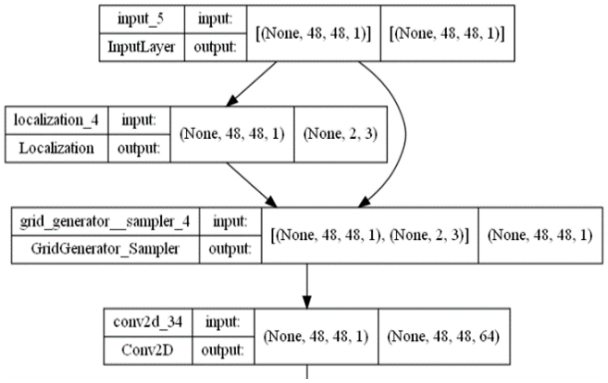

Figure 7: Flow chart about Spatial Transformer layers

Transfer Learning Transfer learning method focuses on utilize the knowledge which was previously gained while solving the different problems and applying it to a targeted task (He et al., 2016). An Increased volume of experiments have investigated pretrained models in the presence of finite learning samples (Sermanet et al., 2013).

Transfer learning leds to faster convergence and outperformed training from scratch ad reduces the overall computational cost because of pre-trained ConvNets (Tajbakhsh et al., 2016). Following figure Figure 8 shows the processing steps on targeted dataset using deep pe-trained model:

For transfer learning models we worked on 4 models where we first define our own parameters and merged them with 4 different selected deep pre-trained models. We selected VGG16, VGG19, InceptionV3 and ResNet50 to detect facial emotion, age and gender. All the parameters for different datasets are explained below:

1. Common Parameters used for all models on all datasets:
   (a) 80% training and 20% testing data.
   (b) Rotation, zoom, horizontal flip for augmentation
   (c) Used 'relu' activation function followed by 'softmax' activation function in last step.

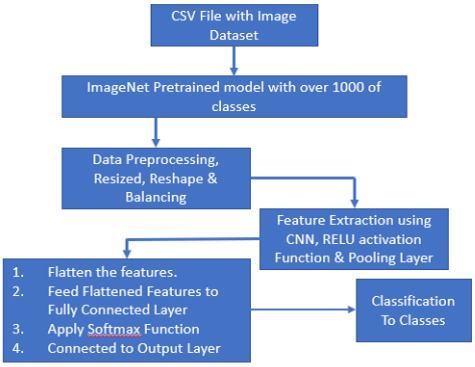

Figure 8: Transfer Learning Method

2. Varying Parameters
   (a) CK+ dataset was trained with (128*128) size and batch size of 7 followed by 100 epocs.
   (b) FER dataset was trained with (64*64) size and batch size of 64 followed by 50 epocs.
   (c) UTK dataset was trained with (224*224) size and batch size of 64 followed by 60 epocs.

Finally, we then compared every model in order to finalize the model based on the validation accuracy and validation loss that we obtained. All the architectures are explained below in detail form.

VGG16 VGG16 is a CNN that is widely regarded as one of the best computer vision models available today (Simonyan et al., 2014). The authors creating this model assessed the networks and increased the depth with a tiny (3x3) convolution filter design, which demonstrated a considerable improvement over prior method. They increased the depth to 16-19 weight layers, resulting in around 138 trainable parameters. VGG16 is an object identification and classification method that can classify 1000 photos from 1000 different categories with an accuracy of 92.7%. It is a common picture classification technique that works well with transfer learning algorithms.

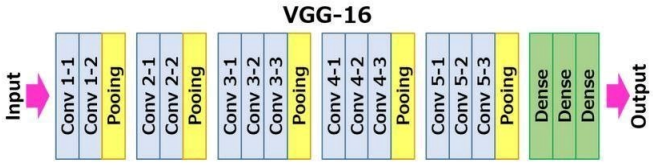

Figure 9: VGG16

VGG16 comprises of thirteen convolutional layers, five Max Pooling layers, and three Dense layers in total, 21 layers, but it only has sixteen weighted layers (i.e. learnable layers) as showed in figure Figure 9 (Simonyan et al., 2014). The input tensor size for VGG16 is 224, 244 with three RGB channels. The most distinctive feature of VGG16 is that, rather than having a huge number of hyper-parameters, they concentrated on having convolution layers of 3x3 filter with stride of 1 and always utilized the same padding and maxpool layer of 2x2 filter with stride of 2. The convolution and max pool layers are positioned similarly throughout the design. Conv-1 Layer has 64 filters, Conv-2 Layer has 128 filters, Conv-3 Layer has 256 filters, and Conv 4 and Conv 5 Layers contains 512 filters. Following a stack of convolutional layers, three Fully Connected (FC) layers are added: the first two layers contains 4096 channels, while the third performs 1000 ILSVRC classification and so has 1000 channels (one for each class). The soft-max layer is the last layer of VGG16.

VGG19 VGG19 is proposed in the same research as VGG16, but it has 19 layers (16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer) as showed in figure Figure 10 (Simonyan et al., 2014). The input tensor size for VGG19 is 224, 244 with three RGB channels. VGG-19 is beneficial because of its simplicity, with 3x3 convolutional layers mounted on top to increase depth level. In VGG-19, max pooling layers were utilized as a handler to minimize volume size. Conv-1 Layer has 64 filters, Conv-2 Layer has 128 filters, Conv-3 Layer has 256 filters, and Conv 4 and Conv 5 Layers contains 512 filters. Similar to VGG16, the last layer of VGG19 is also the soft-max layer.

Figure 10: VGG19

InceptionV3 (Lin et al. 2013) proposed inceptionV3 network, which is a deep neural network with an architectural design made up of repeated components known as Inception modules as shown in figure Figure 11 and Figure 12 An InceptionV3 model's In-

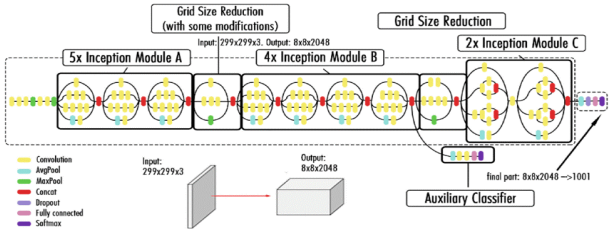

Figure 11: InceptionV3

ception Module works on: Input layer 1x1 convolution layer 3x3 convolution layer 5x5 convolution layer Max pooling layer Concatenation layer Pooling downscales the input data and create a smaller output with a reduced height and width as shown in figure Figure 12. Within an Inception module, padding is added to the max-pooling layer to ensure that it maintains the height and width as the other outputs of the convolutional layers within the same Inception module. The naive inception module enables the use of multiple convolutional filter sizes to learn spatial patterns at different scales, thus increasing the computational cost The computational cost is determined by the mathematical calculations performed within the convolutional layers. The computational cost of naive inception model can be too high to gain any practical benefits from the representational power.

ResNet50 ResNet-50 model is divided into five stages, each comprising a convolution and an identity block. Each convolution block contains three convolution layers, as do the identity blocks. Over 23 million trainable parameters are available in the ResNet-50 (He et al., 2016).

The ResNet-50 Model's modules layes is explained in detail form with figure Figure 13.

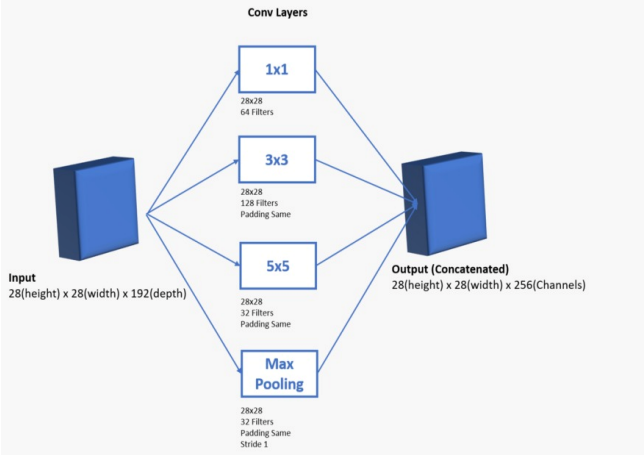

Figure 12: InceptionV3 Modules

1. 1 layer with kernel size of 7x7 and 64 kernels with stride of 2 With output size of 112x112.
2. 2nd a 3x3 max pool with stride 2.
3. In the next convolution, there is a 1x1,64 kernel, followed by a 3x3,64 kernel, and finally a 1x1,256 kernel. These three layers are repeated three times in total, giving us 9 layers in this step with output size of 56x56.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 13: ResNet50 Model Architecture

4. After that a kernel of 3x3,128 followed by a kernel of 1x1,512 this step is repeated 4 time thus giving 12 layers in this step of output size 28x28.
5. It is followed by a kernel of 1x1,256 and two more kernels of 3x3,256 and a kernel of 1x1,1024, repeated 6 time giving 18 layers of 14x14 output size.
6. The second last step consists of a 1x1,512 kernel followed by two, 3x3,512 and a 1x1,2048 kernel, repeated 3 times giving a total of 9 layers of output 7x7.
7. Finally in the last layer we do average pool and complete with a fully connected layer of 1000 nodes and a softmax function giving us output size of 1x1.

Thus, adding all the layers in the above steps 1 + 9 + 12 + 18 + 9 + 1 = we get total 50 layers as shown in figure autoreffig:resnet50, (He et al. 2016) explained.

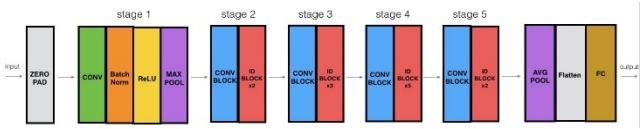

Figure 14: ResNet50

Result and Discussion

As we can observe from Figure 15 and Figure 16, it is clear that transfer learning showed the consistency in both training and validation accuracy throughout the training period whereas ST-CNN had many variations in validation accuracy. But, in the last part of the training, we noticed that both models had very much equal validation accuracy of about whereas there was a major difference in validation loss. All the validation accuracy and loss for the CK+48 dataset has been listed within table Table 3.

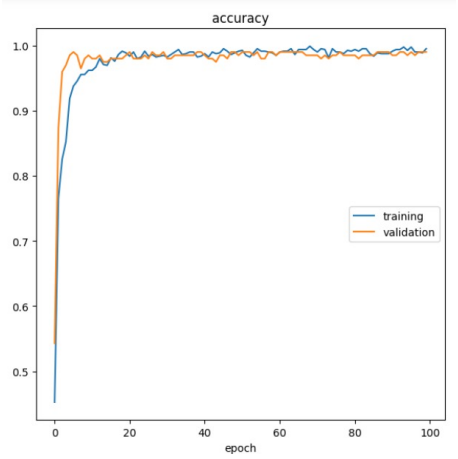

Figure 15: VGG16 Accuracy for CK Emotion

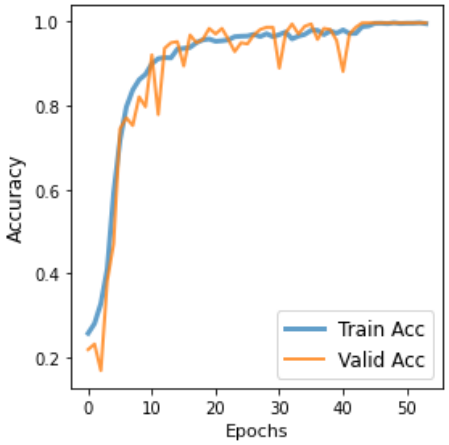

Figure 16: ST-CNN Accuracy for CK-Emotion

On the other hand if we talk about FER2013 dataset, we can state that ST-CNN model had the best training and validation accuracy as compare to all the other models. From the figure Figure 17 we

Model	Accuracy	Loss	F1
Vanilla CNN	0.97	0.14	0.97
ST-CNN	0.98	1.18	0.98
Inception V3	0.95	1.50	0.94
VGG16	0.98	0.04	0.99
VGG19	0.97	0.05	0.98
ResNet50	0.89	0.40	0.85

Table 3: CK+ Dataset Results

can see that even ST-CNN is the best model it had a major difference between training and validation accuracy. All the validation accuracy and loss for the FER2013 dataset has been listed within table Table 4.

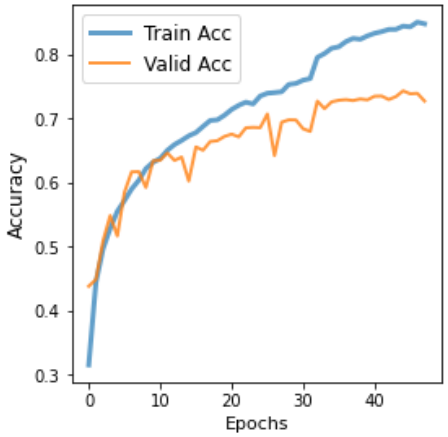

Figure 17: ST-CNN Accuracy for FER-Emotion

Model	Accuracy	Loss	F1
Vanilla CNN	0.62	2.88	0.62
ST-CNN	0.73	1.18	0.73
VGG16	0.52	1.50	0.51
VGG19	0.49	1.56	0.48
ResNet50	0.44	1.53	0.38

Table 4: FER2013 Dataset Results

Finally, for UTK dataset we observed that VGG16 in gender and VGG19 in age provide the best validation accuracy and loss. It can be seen clearly from the figure Figure 18 and figure Figure 19, even if both model had major difference between validation accuracy and loss they were able to classify live facial gender and age at a very acceptable rate. All the validation accuracy and loss for the UTK dataset has been listed within table Table 5.

Conclusion

We conclude that we had insufficient and imbalance data. This made us hard to train an image-based model more effectively with less validation loss and with more sufficient accuracy. For transfer learning methods, training a image-based models with a particular

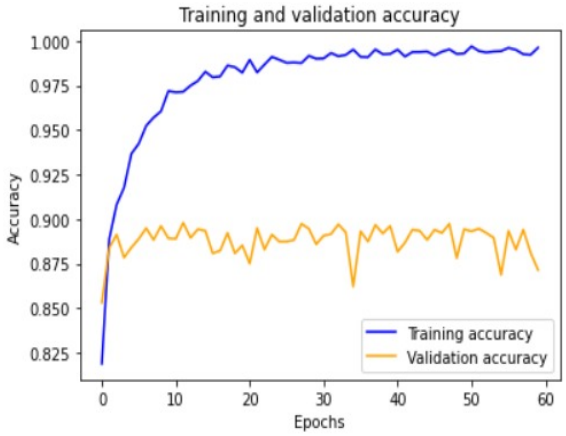

Figure 18: VGG16 Accuracy for Gender

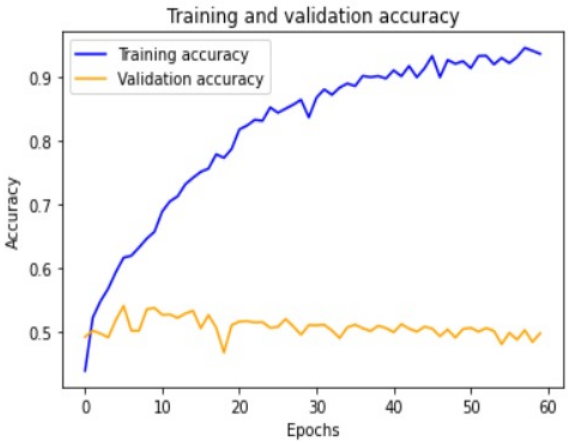

Figure 19: VGG19 Accuracy for Age

size of (224*224) requires more GPU power in order to train models fast and effectively.
Finally, based on the results we conclude that, ST-CNN outperforms in FER2013 database for emotion detection with 0.73 validation accuracy, whereas, VGG19 and VGG16 a transfer learning methods outperforms in UTK dataset for age and gender detection with 0.66 and 0.91 validation accuracy respectively. Lastly, ST-CNN and Transfer Learning method VGG16 performed equally on validation accuracy for CK dataset for emotion detection with 0.98 and 0.98 respectively, whereas, Vanilla CNN performed average in all the datasets.

Future Work

Use Vision Transformer such as Swin Transformers. Incorporate image segmentation techniques into image preprocessing and balancing the data more effectively using various techniques like oversampling to reduce the problem of over-fitting. To use more efficient image dataset to improve the emotion detection accuracy. To work on more deep models to reduce the loss and improve the efficiency of the face features detection. Lastly, also to Include various other factors like ”race” detection.

Model	Accuracy	Loss	F1
ST-CNN (Age)	0.59	1.17	0.56
VGG16 (Age)	0.61	1.05	0.56
VGG19 (Age)	0.66	1.07	0.55
Inception V3 (Age)	0.52	1.30	0.48
ResNet50 (Age)	0.14	1.76	0.04
ST-CNN (Gender)	0.91	0.53	0.90
VGG16 (Gender)	0.91	0.33	0.91
VGG19 (Gender)	0.89	0.51	0.89
Inception V3 (Gender)	0.81	0.38	0.81
ResNet50 (Gender)	0.50	0.70	0.33

Table 5: UTKFace Dataset Results

Live Demo

Figure 20 and Figure 21 are the live demo which was developed using Transfer Learning models. The models of this demo were able to provide very acceptable and sufficient results where it can be seen that top two labels were showing the emotions results from two different models and bottom two were showing the gender and age.

On the other hand Figure 22 is showing the results that were produce by the ST-CNN model where we have provide the same structure for displaying the results as above. We included both Emotion models that are FER2013 and CK+48 to display the live comparison between each other.

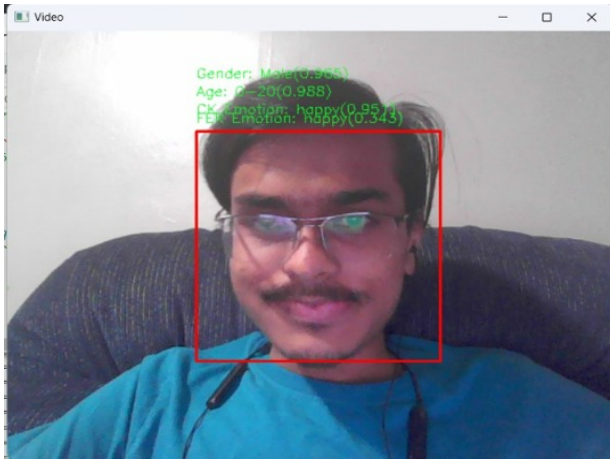

Figure 20: Live Demo 1

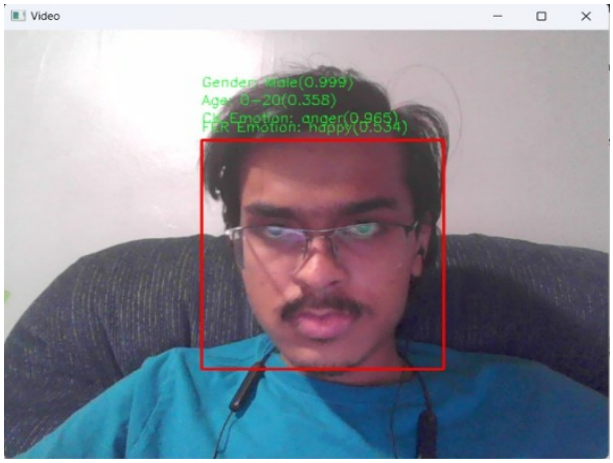

Figure 21: Live Demo 2

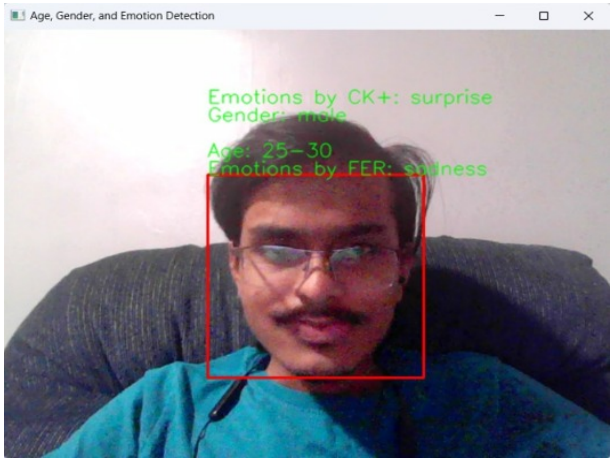

Figure 22: Live Demo 3

Reference

1. Mehrabian, A. (2008). Communication without words. Communication theory, 6, 193-200.
2. Kaulard, K., Cunningham, D., Bulthoff, H., Wallraven, C. (2012). ¨ The MPI Facial Expression Database — A Validated Database of Emotional and Conversational Facial Expressions. Plos ONE, 7(3), e32321. https://doi.org/10.1371/journal.pone.0032321
3. Singh, S., Nasoz, F. (2020). Facial expression recognition with convolutional neural networks. 2020 10th Annual Computing and Communication Workshop and Conference (CCWC). https://doi.org/10.1109/ccwc47524.2020.9031283
4. Minaee, S., Minaei, M., Abdolrashidi, A. (2021). Deep-emotion: Facial expression recognition using attentional convolutional network. Sensors, 21(9), 3046. https://doi.org/10.3390/s21093046
5. Xu, M., Cheng, W., Zhao, Q., Ma, L., Xu, F. (2015). Facial expression recognition based on transfer learning from deep convolutional networks. 2015 11th International Conference on Natural Computation (ICNC). https://doi.org/10.1109/icnc.2015.7378076
6. Ng, H. W., Nguyen, V. D., Vonikakis, V., Winkler, S. (2015, November). Deep learning for emotion recognition on small datasets using transfer learning. In Proceedings of the 2015 ACM on international conference on multimodal interaction (pp. 443-449).
7. Xue, F., Wang, Q., Guo, G. (2021). Transfer: Learning relation-aware facial expression representations with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3601-3610).
8. Li, Y., Zeng, J., Shan, S., Chen, X. (2018). Occlusion aware facial expression recognition using CNN with attention mechanism. IEEE Transactions on Image Processing, 28(5), 2439-2450.
9. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
10. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229.
11. Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning?. IEEE transactions on medical imaging, 35(5), 1299-1312.
12. Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
13. Lin, M., Chen, Q., Yan, S. (2013). Network in network. arXiv preprint arXiv:1312.4400.
14. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).