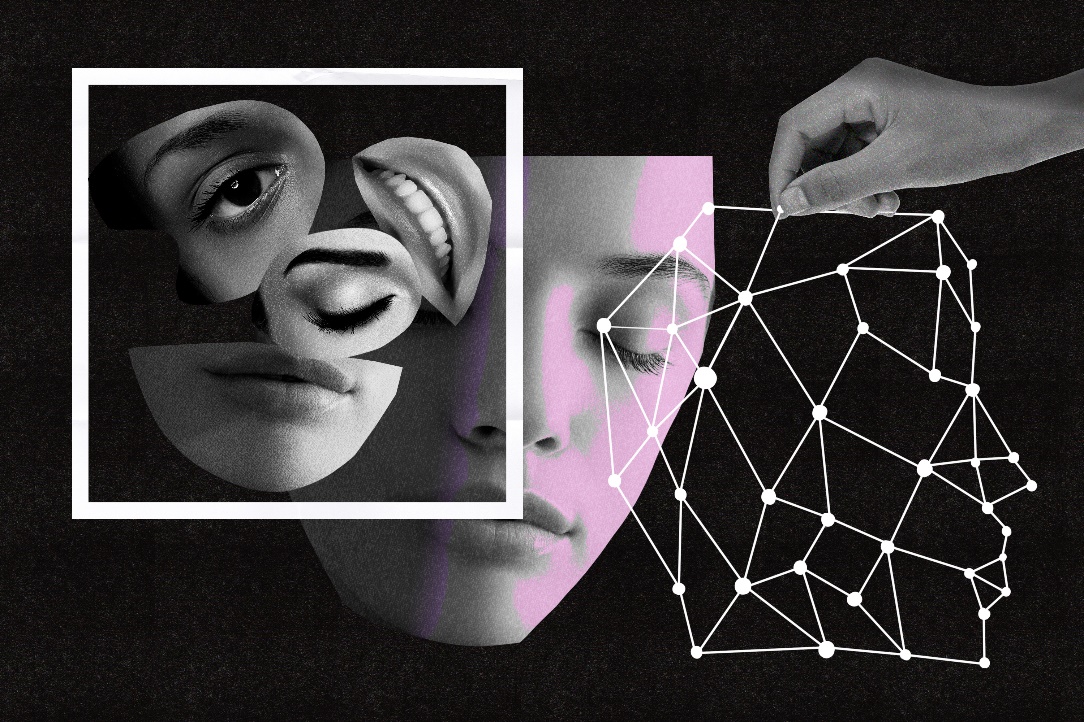


**Software Engineering department**

**Braude Collage**

**Capstone project part B**

## Recognize human emotions from video



**project num:22-2-R-21**

**EDEN ATIAS -316325968**

**edenatiasss@gmail.com**

**EYAL AZRAD-204188783**

**eyalazrad777@gmail.com**

**Supervisor(s):**

**Mr. Ilya Zeldner**

1. Abstract………………………………………………………….3
2. Introduction………………………………………………………3-4
3. Background &Related work…………………………………....4-5
4. Expected Achievements ………………………………………...5
5. Research Process ………………………………………………..5-12
   1. Datasets……………………………………………………...5-7.
   2. DenseNet201……. …………………………………………...7-10
   3. CNN……………………………………………………………23
6. Product…………………………………………………………...12-22
7. Evaluation /Verification plan……………………………………22-23
8. References……………………………………………………….23

# ABSTRACT

Emotion is a mental-physical state that has subjective and objective expressions. It may be expressed in different ways. Emotion recognition is a developing field, which may be useful in many fields. We believe that our project, which aims to identify emotions in a video where humans are present, may contribute to many fields such as: media, industry, various technologies, etc. Therefore, our challenge is to build a system that will be able to correctly classify human emotion from a video that he takes part in. In this paper we will describe our project, we are proposing a deep learning-based approach to recognizing the human mental state. We are working with six universal emotions (happiness, sadness, fear, anger, neutral, and surprise) and with a data set that we will create that contains 2 features: eyes and their surrounding areas and lips. The method that we are using in our project consists of feature extraction, feature selection, and classification. To extract the relevant features, we will use Media –pipe library, and then we will use 2 pre-trained neural network models in order to classify the emotions from the eyes and the lips. For the eyes, we will use the architecture of DenseNet201 and for the lips, we will use CNN. Our algorithm consists of 2 parts that work in parallel and in the end, we will combine the results of the two models and we will detect the emotion according to the emotion that received the highest prediction score for the frame section (the was executed from the video) we are testing. We built a method for classifying the result that takes into account the most common result classified by the 2 models.

***Keywords***—Human emotion recognition; convolutional neural network (CNN); Media-pipe; features; freams; detection; DenseNet-201

# introduction

Expressions are signs of feelings and enable a person to communicate their current emotional state. Understanding emotional cues is essential for recognizing affective behavior of humans. In this essay, we investigate how to identify human emotion by identifying observable changes in the eyes and the mouth. The importance of emotion recognition from facial expressions is enormous, especially for frameworks for human-machine interaction. Numerous studies have been conducted to build and create systems for recognizing facial emotions, however, the emotion recognition process still has a number of common issues. First, it is found that the characteristics are quite sensitive to variations in noise, light, and occlusion. This suggests that a small change in noise, lighting, or occlusion may lower the accuracy rate of the recognition process, and it also suggests that the performance of such systems is influenced by the size of the data set. The problem of categorizing and recognizing objects has advanced due to the extraordinary development of deep neural networks and convolutional neural networks as well as the availability of the necessary data. Many intricate recognition tasks were assumed to be difficult and demonstrate lower accuracy. Higher accuracy can be attained with the use of convolutional neural networks (CNN). Today, the majority of research focuses on identifying emotions from a single facial feature. In order to determine emotions from the eyes and lips individually and combine the data, we want to develop an integrated algorithm. One of our main goals is that our project will assist people with disabilities that are dealing with difficulties in identifying others emotions.

# Background

**Convolutional neural networks (CNNs**): a type of neural network frequently employed in the processing of images and videos. They are designed to analyze data having a topology that resembles a grid, like an image, where features at the grid's edges have a stronger relationship to features nearby than to features further away. Convolutional layers, which CNNs use, apply a series of filters to the input data and train themselves to detect particular features. The filters compute dot products between their entries and the incoming data as they glide over it, creating a 2D activation map. To extract higher-level characteristics and produce a prediction, this map is subsequently processed by one or more additional layers, such as pooling and fully connected layers. They are frequently employed in tasks involving object and picture classification.

**Media Pipe**: an open-source framework developed by Google for building cross-platform multimodal (e.g. video, audio, text) applied ML pipelines. It offers a set of reusable machine learning and media processing parts that can be combined to build unique pipelines for tasks like object detection, face landmark detection, pose estimation, and stylization, among other things. A number of pre-built solutions for typical use cases, like hand and face tracking, are also included in MediaPipe and may be quickly integrated into a pipeline. With support for a number of input and output formats and the ability to run on a variety of platforms, including mobile devices, the framework is made to be adaptable and simple to use.

# Related work

1. **Recognizing Human Emotions from Eyes and Surrounding Features: A Deep Learning Approach:**Shuvo, M. N. R., Akter, S., Islam, M. A., Hasan, S., Shamsojjaman, M., & Khatun, T. (2021). Recognizing human emotions from eyes and surrounding features: a deep learning approach. *International Journal of Advanced Computer Science and Applications*, *12*(3).‏
2. **Four‑layer ConvNet to facial emotion recognition with minimal epochs and the signifcance of data**: diversityDebnath, T., Reza, M., Rahman, A., Beheshti, A., Band, S. S., & Alinejad-Rokny, H. (2022). Four-layer ConvNet to facial emotion recognition with minimal epochs and the significance of data diversity. *Scientific Reports*, *12*(1), 1-18.‏
3. **Building Emotional Machines: Recognizing Image Emotions through Deep Neural Networks:** Kim, H. R., Kim, Y. S., Kim, S. J., & Lee, I. K. (2018). Building emotional machines: Recognizing image emotions through deep neural networks. IEEE Transactions on Multimedia, 20(11), 2980-2992.‏
4. **Enhancing Mouth-Based Emotion Recognition Using Transfer Learning :** Franzoni, V., Biondi, G., Perri, D., & Gervasi, O. (2020). Enhancing mouth-based emotion recognition using transfer learning. *Sensors*, *20*(18), 5222.‏

# Expected achievements

This project involves creating software that will take as input a video and will know how to classify the feelings of the person who appears in the video at any given moment. Given the fact that a video consists of several frames, we capture every few moments an image that is sent to our pre-trained network. The emotions that will be identified are six universal emotions -happiness, neutral, sadness, fear, anger, and surprise. To achieve our goal, we will create a CNN –architecture based network and train it by a wide variety of images from a dataset consists unique double eye images we built. To get more precise results we also create another network that will detect human emotions from their lips. We are will analyze the video frame by frame, in each frame we will extract the face area, following that we will find the position of the mouth and the eyes and their surrounding areas. The system will help autistic people to recognize different emotions.

* Build a CNN network with all the sublayers correctly.
* Extract the relevant face features that will help to categorize emotions.

One of the challenges we will face is extracting the relevant features from the video and building a neural network that can learn and recognize human emotions with success. Another challenge we face is to combine the results from both of the algorithms that we are building (one to detect emotions from the eyes and the other from the lips).

# Research

**DATA- SET:**

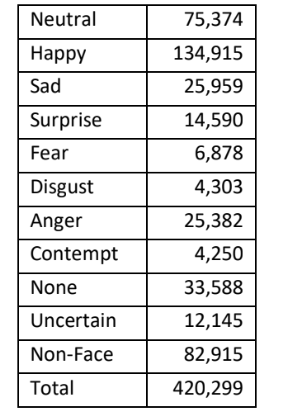
**Database AffectNet**

* About 1 million facial photos were gathered for AffectNet from the Internet using three of the most popular search engines and 1250 emotion-related keywords.
* AffectNet provides:

1. Images of the faces
2. Location of the faces in the images
3. Eleven categorization classifications for emotions and non-emotions (Neutral, Happy, Sad, Surprise, Fear, Disgust, Anger, Contempt, None, Uncertain, No-Face)
4. **Emotion categories:**

0: Neutral, 1: Happiness, 2: Sadness, 3: Surprise, 4: Fear, 5: Disgust, 6: Anger,

7: Contempt, 8: None, 9: Uncertain, 10: No-Face



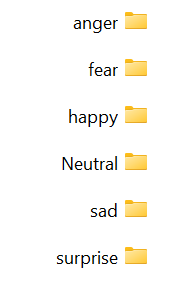


In our project, we are classifying 5 main emotions – Happy, Sad, Surprise, Fear and Anger. we used the images that belong to these emotions in the date set. We also used images belonging to a neutral emotion, which will be a control group for the lack of emotion in the person's facial expression.Since we are working with a method of supervised learning, we needed labeled data. As was mentioned before our data is labeled and we created in our code function that created a train data folder and 6 folders inside for each emotion (that contain the same label). The key to an accurate emotion prediction from our entire model is that we are using the same data –set and after each architecture is classifying an emotion, we are making sure that the different face features each architecture received belong to the same person. Since the data set contains face images we used media –pipe library to extract each face feature (eyes and their surroundings, lips). We received a labeled feature data set divided into files by emotions.





**\*On the left: the image of the face as it is in the data set, on the right the face features that were cut using the media pipe library and were send as an inputs to our models.**



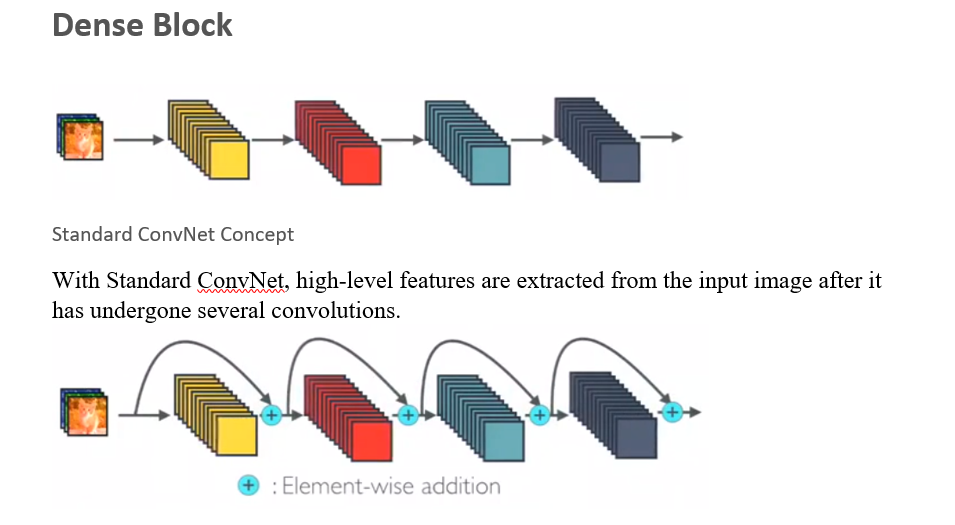
**\*Dividing the images into folders by emotion**

**Architectures:**

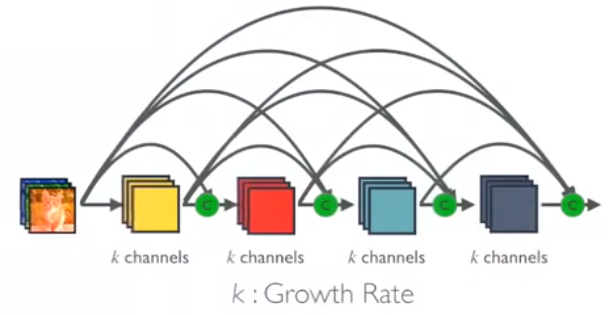
We used 2 CNN-based architectures, one for each model. **For the task of detecting emotion from the eyes and their surroundings:**

**DenseNet 201**

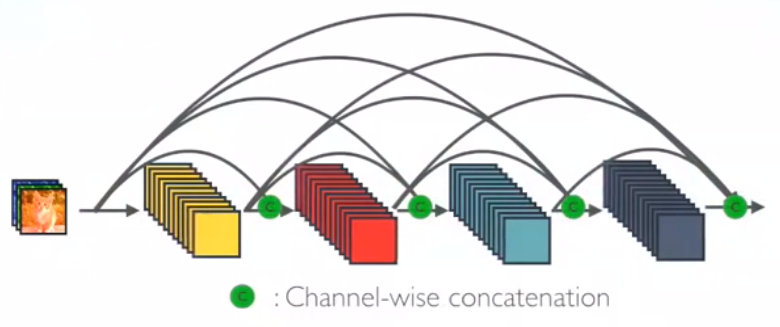
A difficulty with the convolutional neural network's increased depth was that as it passed through more layers, information about the input or gradient would disappear. Dense Net authors developed an architecture with a straightforward connectivity pattern to address this issue, ensuring the greatest possible information flow between layers for both forward computation and reverse gradient calculation. This network connects all layers in a way that allows each layer to receive additional input from all layers that came before it and to pass its own feature maps to layers that came after it. There are L layers in the network. The non-linear transformation Hl([x0,x1,...,xl1])Hl([x0,x1,...,xl1]) is implemented by each layer, where l indexes the layer. Hl(.)Hl(.) is a composite function that can include operations like batch normalization (BN), rectified linear unit (ReLU), pooling, or convolution (Conv).



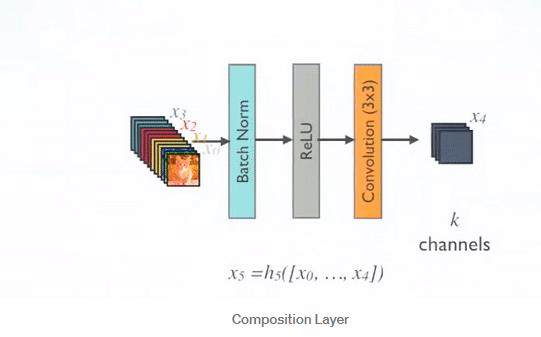


In Dense Net, each layer receives new inputs from all lower layers and transmits its own feature maps to all higher layers. It's done using concatenation. 

**Dense Block in DenseNet with Growth Rate k**

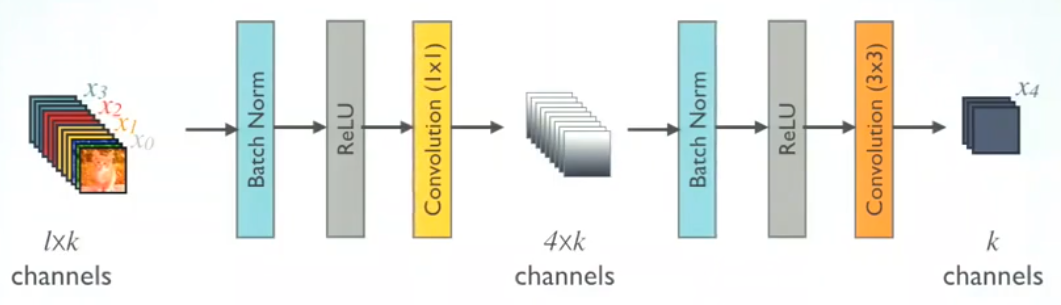
The network can be smaller and more compact, meaning there can be fewer channels, because each layer receives feature maps from all preceding levels. The extra number of channels for each layer is the growth rate k.

**So, it has higher computational efficiency and memory efficiency.**

****

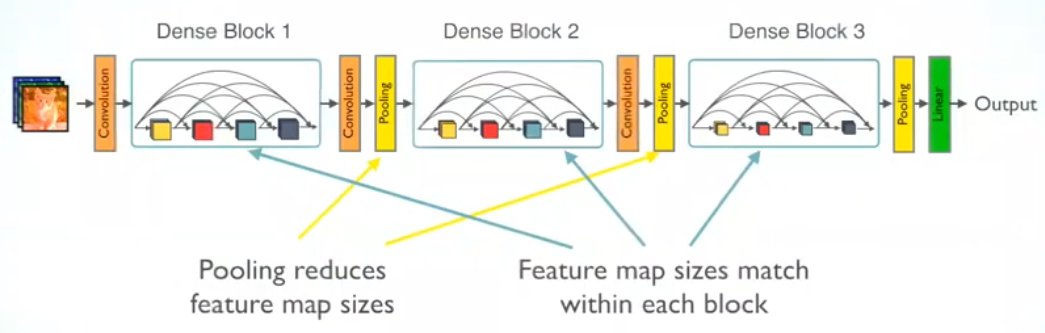
## Pre-Activation Batch Norm (BN), ReLU, and 3X3 Conv are performed for each composition layer with output feature maps of k channels

## DenseNet-B (Bottleneck Layers)



****DenseNet-B

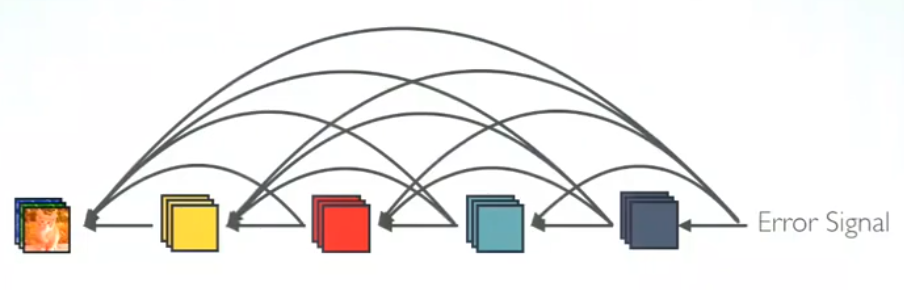
## Multiple Dense Blocks with Transition Layers



The transition layers between two adjacent dense blocks are 1X1 Conv and 2X2 average pooling. Within the dense block, feature map sizes are uniform, making it simple to concatenate them. A softmax classifier is applied once a global average pooling is completed at the conclusion of the final dense block.

# Advantages of DenseNet

## 1. Strong Gradient Flow



## 2."deep supervision" is implied direct propagation of the error signal to prior layers is simple. As earlier layers can receive direct supervision from the final classification layer, this is a form of implicit deep supervision.3.2. Parameter & computational efficiency.

The classifier in DenseNet incorporates features of all degrees of complexity. It typically provides more supple decision boundaries. It also explains why DenseNet functions well in the absence of sufficient training data. One of the models in the DenseNet family created for picture classification is the densenet-201 model. The size and precision of the densenet-121 model are the primary differences. Compared to the densenet-121 model's around 31MB size, the densenet-201 is larger at over 77MB.

**For the task of detecting emotion lips:**

**CNN:**

CNN (Convolutional Neural Network) architecture is a type of deep learning architecture that is primarily used for image and video processing tasks. It is composed of several layers, including:

**1**. The CNN's **input layer** is the top layer and is where the raw picture data is sent.

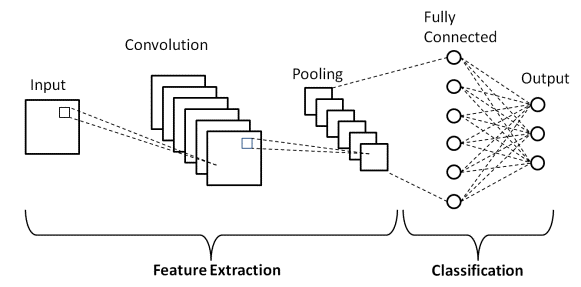
**2. Convolutional Layer:** This layer applies convolution operations to the input image in order to identify various features such as edges, textures, and patterns.

**3. ReLU (Rectified Linear Unit) Layer:** This layer converts the output of the convolutional layer into a non-linear activation function known as ReLU. This layer's goal is to make the network non-linear so that it can learn more intricate representations of the input data.

**4. Pooling Layer:** This layer performs a down-sampling operation on the output of the ReLU layer. The purpose of this layer is to reduce the spatial dimensionality of the output, which helps to reduce the computation required by the network and also make the features more robust.

**5.Fully Connected Layer:** This layer connects all the neurons of the previous layer and it is used to classify the image.

**6.Output Layer:** This is the final layer of the CNN, and it produces the final output of the network, which is the predicted class of the input image.

****

**Measurements tools:**

**Confusion Matrix**

When trying to resolve categorization issues, a confusion matrix is a highly common tool. Both binary classification and multiclass classification issues can be solved with it. Confusion matrices represent counts between expected and observed values.

**Accuracy=**

**The components of the formula:**

**TN** -stands for True Negative which shows the number of negative examples classified accurately.

**TP**-stands for True Positive which indicates the number of positives.

**FP**-shows False Positive value the number of actual negative examples classified as positive.

**FN** -means a False Negative value which is the number of actual positive examples classified as negative.

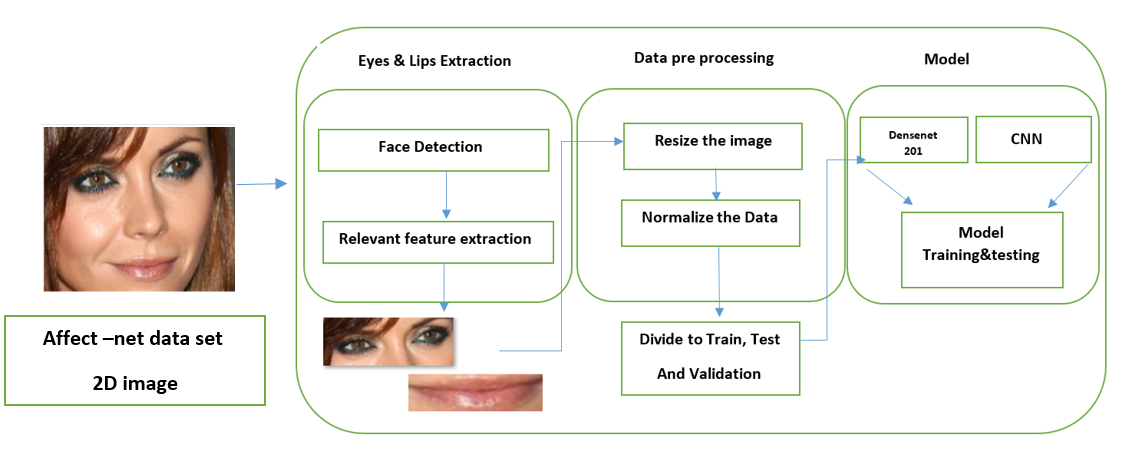
**To obtain a confusion matrix, users need to provide actual values and predicted values to the function.**

# product

**Data pre-processing:**

As was mentioned before we worked with Affect- net data set. But the images it contains is full face images. So we used media –pipe library in order to extract the relevant face features we needed. This is how we created for each architecture unique data –set with the face feature it needed. Before we sent the images we used resize so function so the input images we sent for training was in size 48\*48. We also normalized that data. Normalizing the data before training a model is important because it helps adjust the data to have a similar range of values. This can improve model performance and make the training process converge faster.

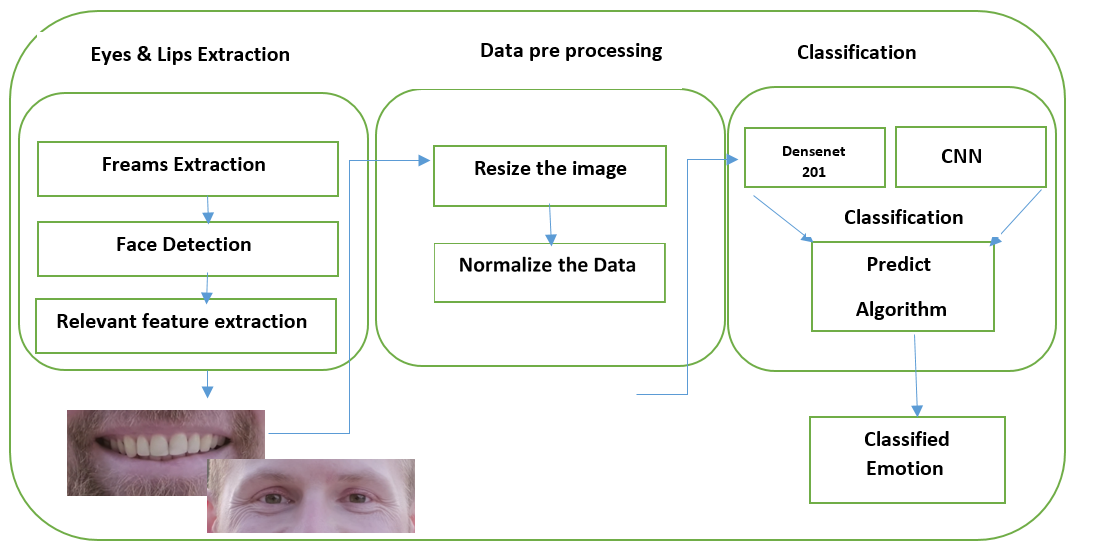
# MODEL DIAGRAMS

**WorkFlow:**

**1.1 Training process:**

****

**1.2 Emotion detection system:**



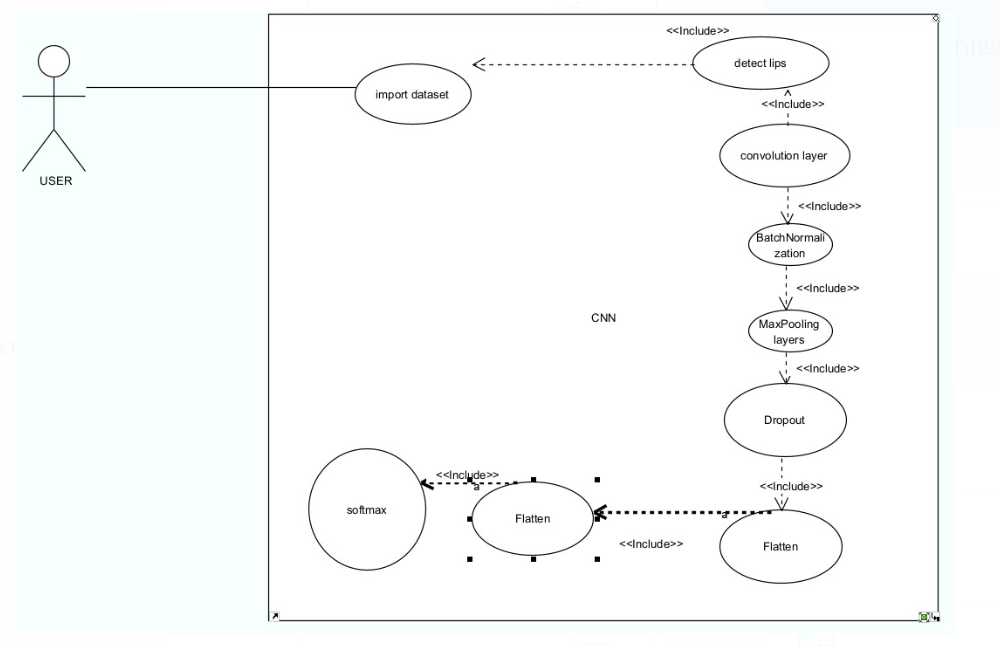
**Video input**



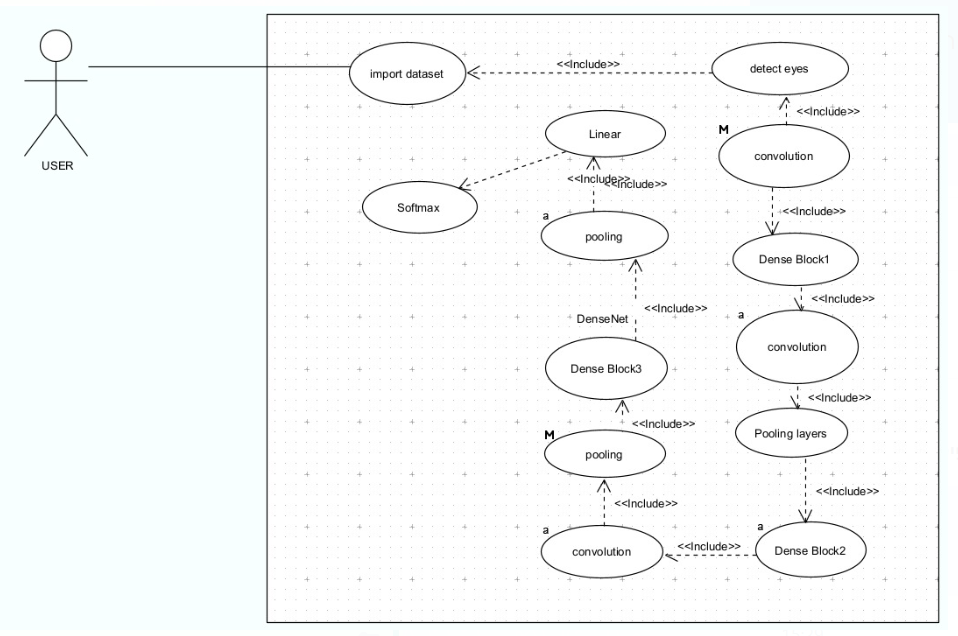


**Model Use case diagram:**

**Training:**

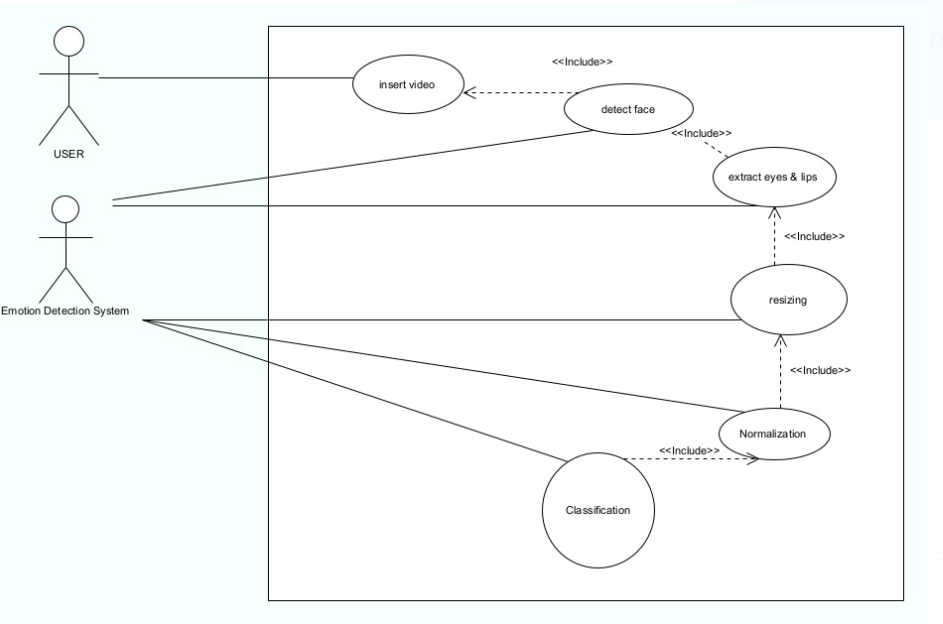
****



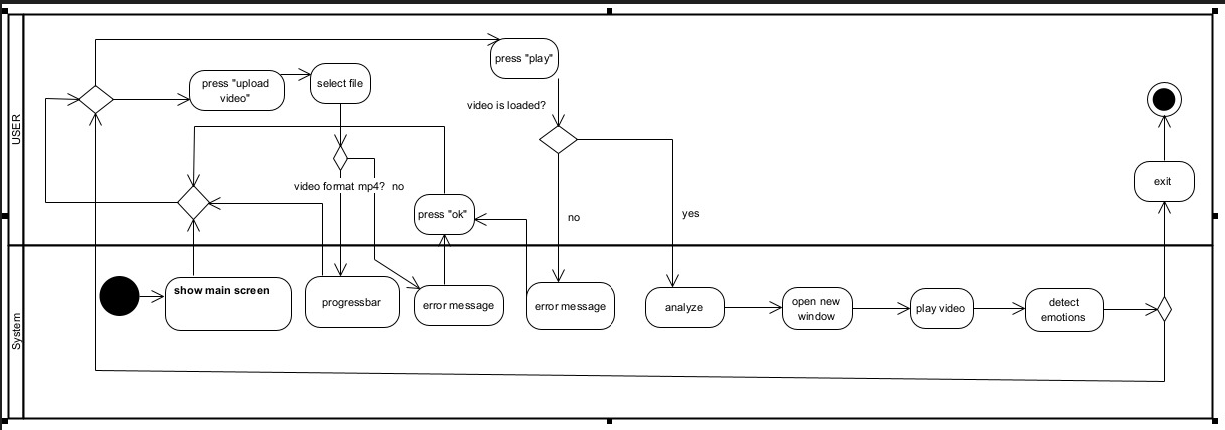


**Emotion Detection System:**

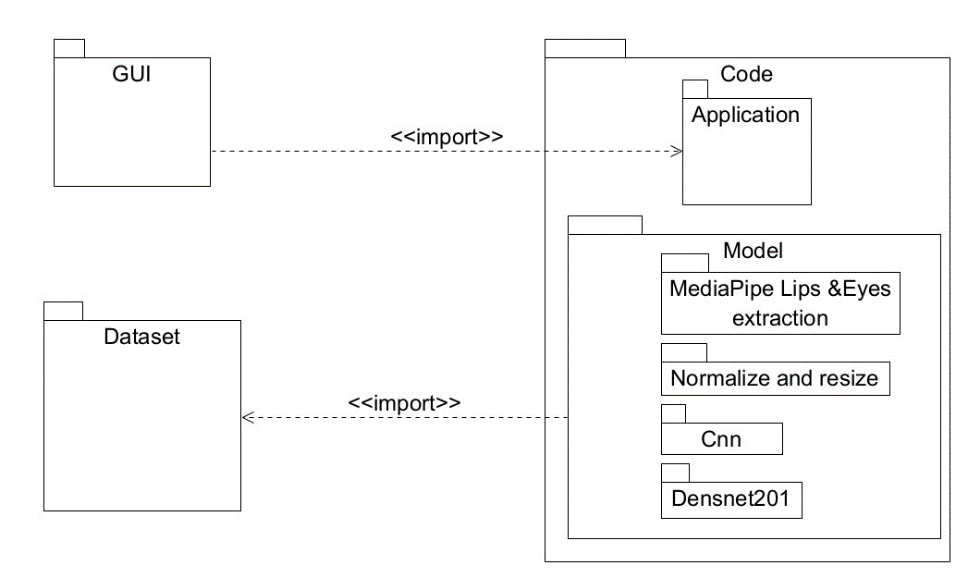
****



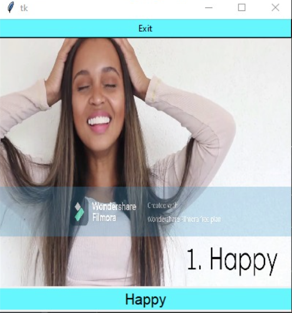
**Activity Diagram:**



**Package:**

****

# Screens

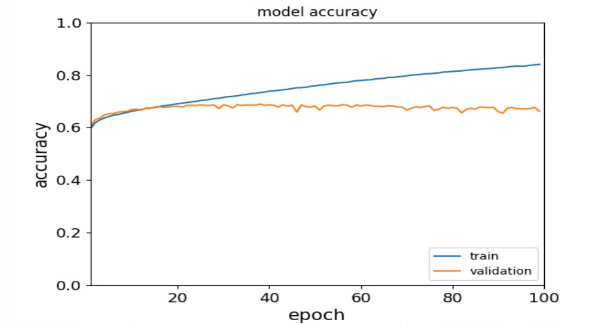


# RESULTS

**Training process:**  our data set contains 200,000 images for each relevant feature. We defined in the code that 80% of this data will be used by the model for training and the 20% left will be used by the model for validation. This is how the model will be able to test his predictions, once it's able to achieve high accuracy the model is ready for use.

**Training Accuracy:**  the most challenging part of our project was to train the model until it will classify the emotions with high accuracy. The challenge was doubled since we trained 2 models, on different face features. We used pretrained models. A pre-trained model is a model that has already been trained on a large dataset and can be used for transfer learning, in which the model's weights are used as a starting point for training on a new dataset with similar characteristics. And yet it took us a while to achieve good accuracy on our model.

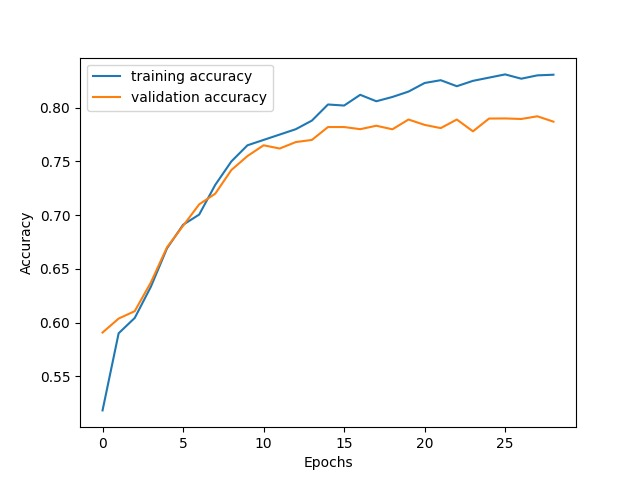
The graphs below will demonstrate the training accuracy we received during our training process, and we will explain the steps we performed in order to receive better results:



**CNN**

**Densenet 201**

We ran the training process on our models, as can be shown in the graphs the validation accuracy increased along with training accuracy until epoch (~20 in cnn and 10 in densenet), and then it reached saturation (need to be mentioned that we ran the training process on about 20k images in this section), after analyzing the graphs we understood that the training reached overfitting from this epoch. Overfitting occurs when a model is too complex and performs well on training data but poorly on new data. This means that the model has learned the noise in the data and not the underlying pattern. Therefore, we understood we need to increase the amount of the training data, and maybe decrease the number of epochs. We tested a few combinations of epochs and data set amounts until we found the best one that provided the best accuracy results. We also performed data augmentation: use techniques like rotation, inversion, cropping, etc. In order to increase the amount of data available to the model, reducing the chance of it overfitting.

**After trying all of the mentioned above we reached this graph:**



As can be shown in the graph we were able to reach Val accuracy of almost 0.8.

**Testing Data results:**

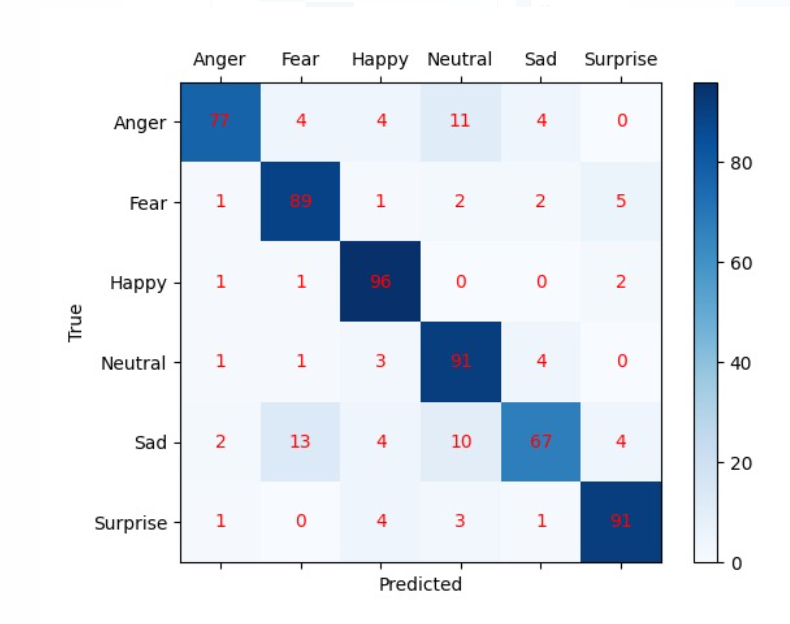
It is important to note that during the training process, we noticed that each facial part behaves differently for different emotions. For example, the differences between angry and neutral expressions will be expressed in a change in the eyes appearance, while the position of the mouth stays pretty much the same in these emotions. And there is no doubt about the emotion that smiling lips represent. This is why we decided to give different weights in the final classification for different emotions that were classified from our models. for example if the CNN model (classifying emotions from lips images)has higher accuracy than the DenseNet model, it will have more weight in our final classification.Question: will you be able to identify which emotions this lips represents?

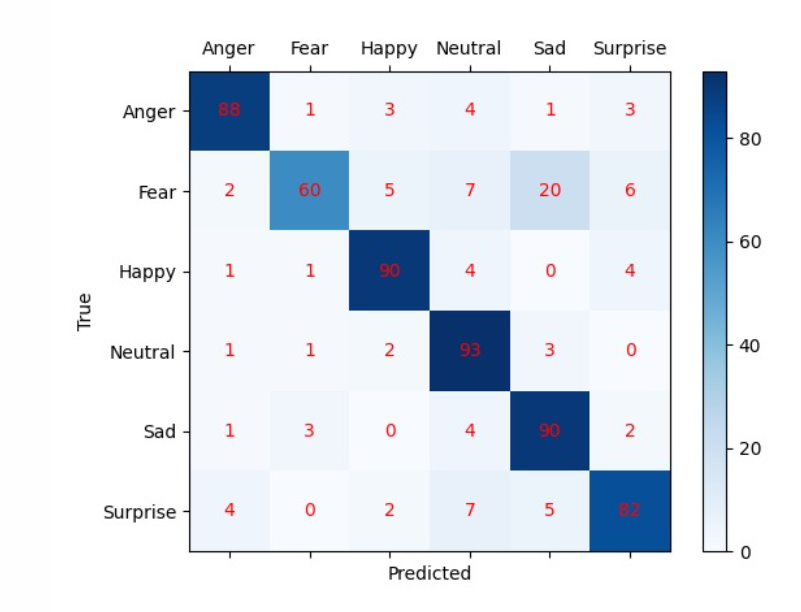
The position of the mouth looks the same but nevertheless, they belong to people who experience different emotions. Here you can see the full images:

Now when we are able to observe all of the facial expression we will be able to identify the left image as a neutral face and the right one as an angry face. The main difference between it reflects in the eyes expression. This is the confusion matrix we received from the CNN model. As was mentioned before CNN classifies the emotion based on lips images:



And this is the confusion matrix from DenseNet model (classifies the emotion based on eyes images):



We received these matrices that represent the results of each model classification that was performed on the testing data, as can be shown both of the modal classify emotions like neutral and happy with high accuracy, from the other hand it seems like Fear is more clearly visible in the changes of the lips patterns, while Sadness is more expressed in the changes of the eyes gaze.

**Final Result:**

Ensemble is a method in machine learning where multiple models are combined to create a more powerful model. The idea behind ensemble methods is that by combining the predictions of multiple models, the resulting ensemble model will be more robust and have better performance than any individual model. For our final result, we will use **weighted Ensemble**. Using this approach, the models are trained independently, and then their predictions are combined by taking a weighted average of the output probabilities, where the weight is determined by how well each model performed. This is why we gave in the final calculation more weight for the CNN classifications of fear and surprised and the Densenet received highest weights for anger and sadness. The system will classify the emotion for a given frame according to the emotion that received the highest accuracy from both models. We analyzed the proportion of success of each model, and weighted them accordingly. For example, for the emotion of fear, the model had a 60% success rate when analyzing the eyes and an 89% success rate when analyzing the lips. Therefore, we gave a higher weight to the lips (89/60 = 1.5) and used that weight in our final ensemble model for the emotion of fear. For emotions that have very small differences, such as happiness, where one model had a 90% success rate and the other had a 96% success rate, we did not assign a weight, as the difference is small and could be influenced by the data set used. Regarding the accuracy of the final model, it is difficult to compare it to the accuracy of the individual models because it operates on videos rather than images. However, the ensemble model achieved an estimated accuracy of 91% based on a small dataset of folders of videos of emotions.

**User Guide:**

**Software Environment:**

* PyChram
* Python 3.8

**Python libraries that needed to be installed in order to execute the code:**

* Keras
* Media –pipe
* Sklearn
* Numpy
* Os
* Matplotlib
* Seaborn
* cv2
* Shutil
* Tkiner

\*The input video must be mp4 type, for now, the system supports only these types of videos.

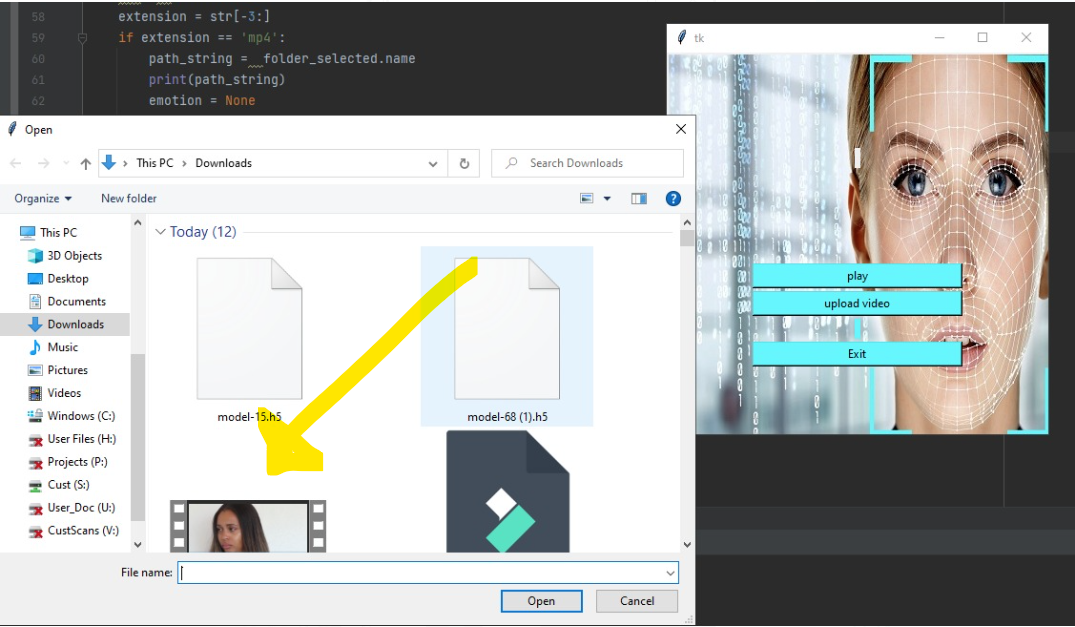
**General Description:** The system is based on deep learning and is designed to classify human emotions from a given video. The system does not classify emotion with 100% accuracy, but can be used as a good base for future development. Every user can train the model by himself, with his own data set, most of the code is designed generically, so every user will be able to adjust his own data paths, size and unique information.

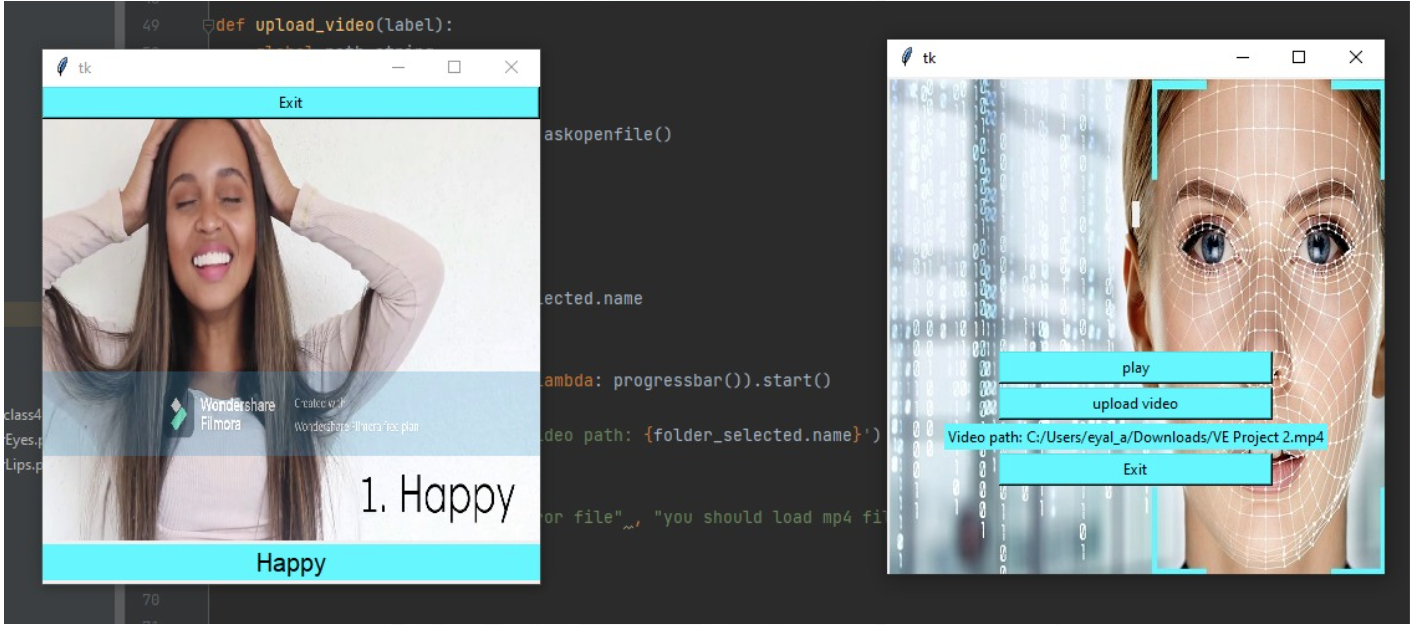
**User instructions:**

**1.** Enter pycharm and go to "GUI" file and press run



2. Press the button "upload video" and choose from file the video you want to analyze



3. press the "play" button

You will see the emotion label during the video on every emotion that was classified by the system.

**Conclusions:**

In this paper, we are proposing a new method for detecting human emotions from video. We use the Media-pipe library for Face detection and in order to extract the eyes and the lips. We use 2 pre-trained models CNN and DenseNet201 for the classification of emotions. Each architecture was trained on different face features, and each classifies an emotion with an overall accuracy of 0.8. Then we are using weighted Esamble in order to classify the input emotion. During the research, it was discovered that different parts of the face, such as the eyes and lips, play a crucial role in recognizing emotions. Different emotions are expressed differently in each part of the face, and as such, several neural network models had to be used to classify emotions from different facial features. To improve the performance of the model, an ensemble model was created that combined the results from the two models to achieve better performance. We believe that with a larger dataset and more advanced computational resources, we can achieve even more accurate results. Our current system has a solid foundation and has the potential for further development.

# EVALUATION & VERIFICATION PLAN

**Simulation:**

|  |  |  |
| --- | --- | --- |
| Expected result | Test Description | Test ID |
| Systems output :on the displayed video the emotion **Happy** will be presented. | Input: frame from a video, of a person that smiles | 1 |
| Systems output :on the displayed video the emotion **Sad** will be presented. | Input: frame from a video, of a person that cries/has sad expression | 2 |
| Systems output : on the displayed video the emotion **Anger** will be presented. | Input: frame from a video, of a person that has angry expression | 3 |
| Systems output : on the displayed video the emotion **Surprised** will be presented. | Input: frame from a video, of a person that has a surprised expression | 4 |
| Systems output : on the displayed video the emotion **Fear** will be presented. | Input: frame from a video, of a person that has feared expression | 5 |
| Systems output : on the displayed video the emotion **Neutral** will be presented. | Input: frame from a video, of a person that has neutral expression | 6 |

**Gui**:

|  |  |  |
| --- | --- | --- |
| Test ID | Test Description | Expected result |
| 1 | Press "upload video" and Insert the expected video file format | The system will display on the main screen label with the path of the video |
| 2 | Insert a wrong video file format | The system will return an Error message: "you should load an mp4 file". |
| 3 | Press the “upload video” button to upload a new video. | The system will override the old video and load the new video instead of |
| 4 | Press the “play” button without inserting the video file | The system will return an Error message: "you should upload a video". |
| 5 | Press the “play” button after inserting the video file | The system will create a new frame that plays the video and analyzes the emotions |

# References

1. Densenet201 - <https://towardsdatascience.com>

2. CNN - <https://www.upgrad.com/blog/basic-cnn-architecture>