COPY ME

Using facial expression

1)Team Name : DYNAMIC

-COPY ME is an Application that teaches facial expression to help Autistic Children.

-We designed two versions to help children get benefits from our model (Mobile Application, Desktop Application)

Each member's role:

- **Dina** : Mobile Application
- **Esraa** : Desktop Application
- **Nourhan** : Model training

2)The Scenario:

- Autistic Children have a problem in expressing their feelings in the correct way so our application helps them in training their face to make the right facial expression by recognizing their faces and comparing it with the photo they are trying to make after choosing it from the application
- It’s important for Autistic Children to learn facial expression to be able to express their feelings right
- Our goal is helping Autistic children to learn facial expression and express their feeling
- Our target group is the autistic children’s parents and the organizations who is in charge of helping them

3)System Demo. Videos:

1. Desktop demo: https://drive.google.com/file/d/18dvJfqVDPZT9vfUn6X-5jjcPJqp6M9Tz/view?usp=sharing
2. Mobile demo: https://drive.google.com/file/d/1AAL5lhjaDvPid_hlhzlJio45a3bPOehj/view?usp=drivesdk

4)Description of the videos:

First : The user starts with choosing what photo he wants to make

Second : He opens his web cam (or mobile cam) and begins to make the expression

Third : The model will recognize his face and the expression he makes ,Then will compare the photo he chose with the expression he made and show his results

The features of the applications:

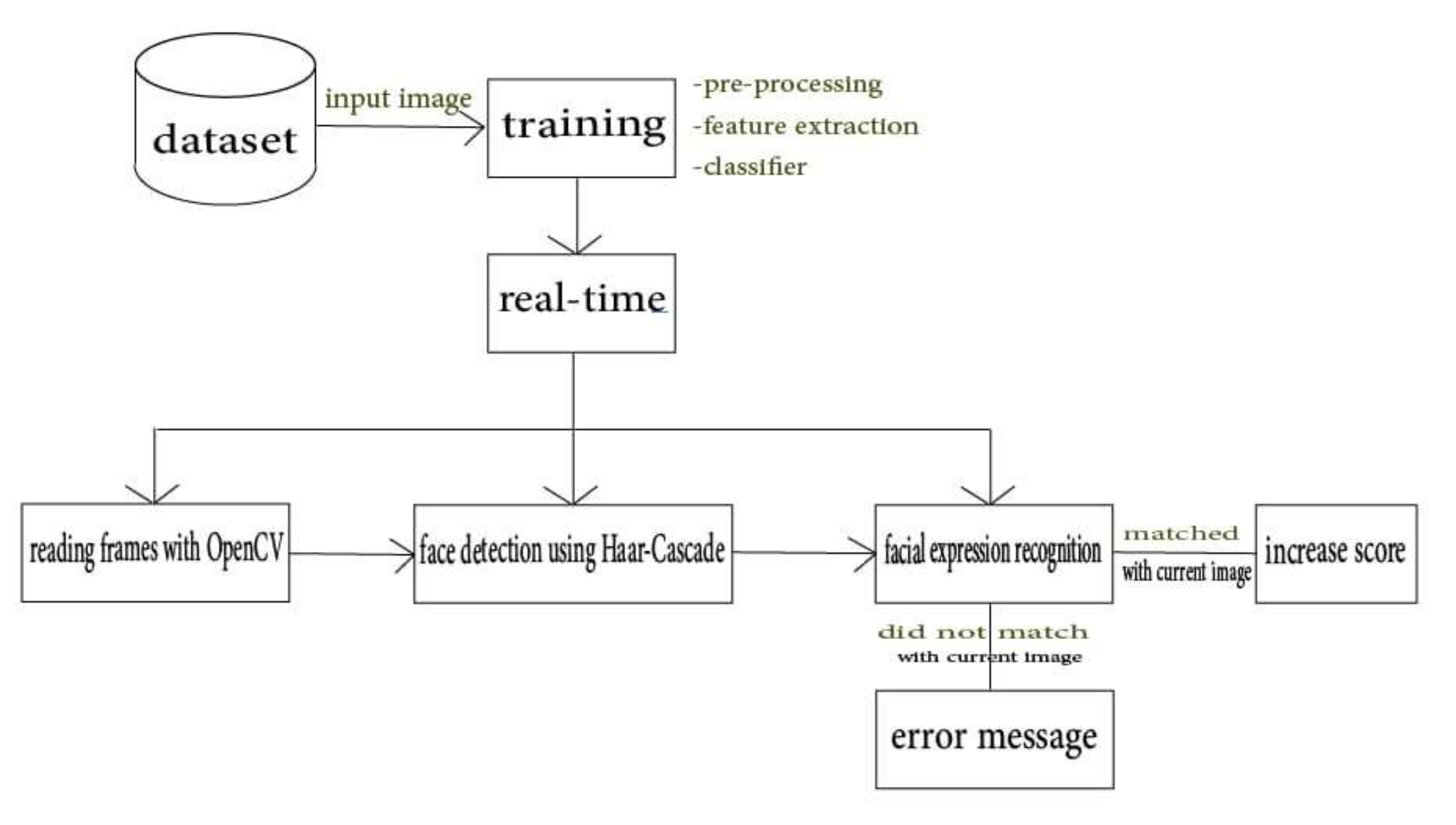
-It teaches autistic children facial expressions so they can express what they feel through their faces

-It allows children to learn emotions with instant feedback on their performance.

-Our app will work like a therapist that follows the development of the condition of autistic children

5)Technical architecture:

General Structure



- **Dataset:**

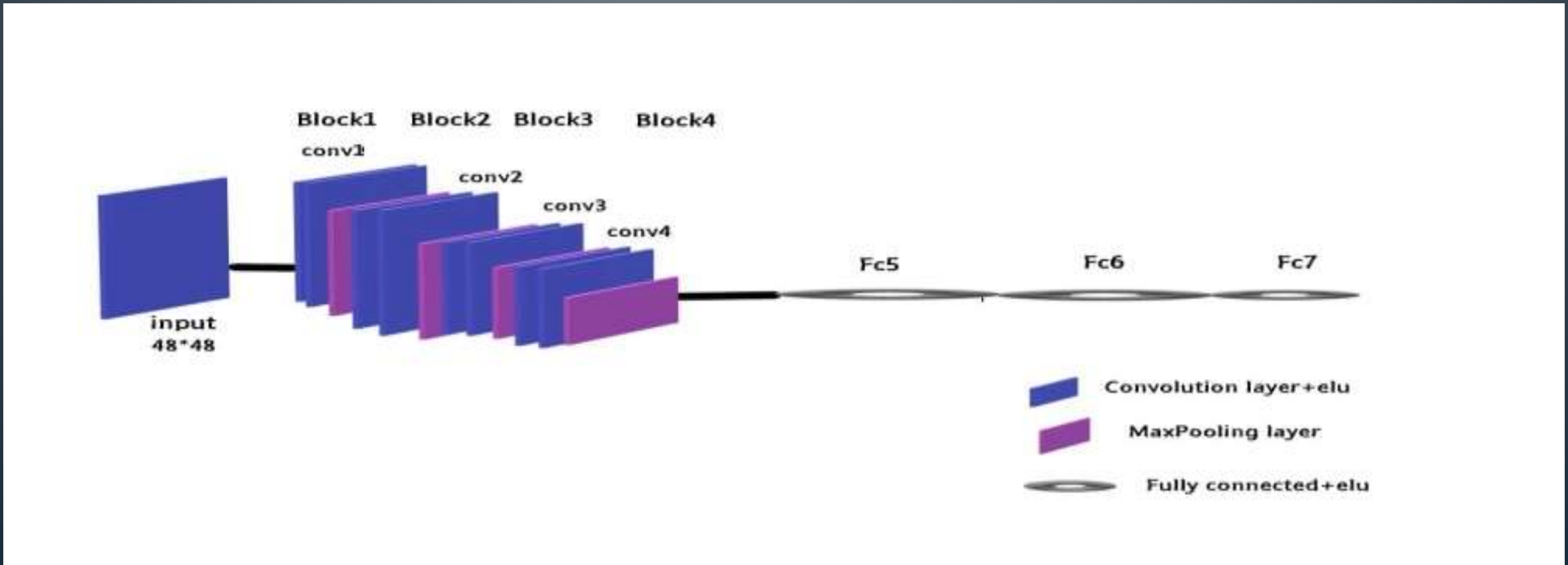
https://www.kaggle.com/msambare/fer2013

40000 images each image has a size 48*48

- **Machine Model :**

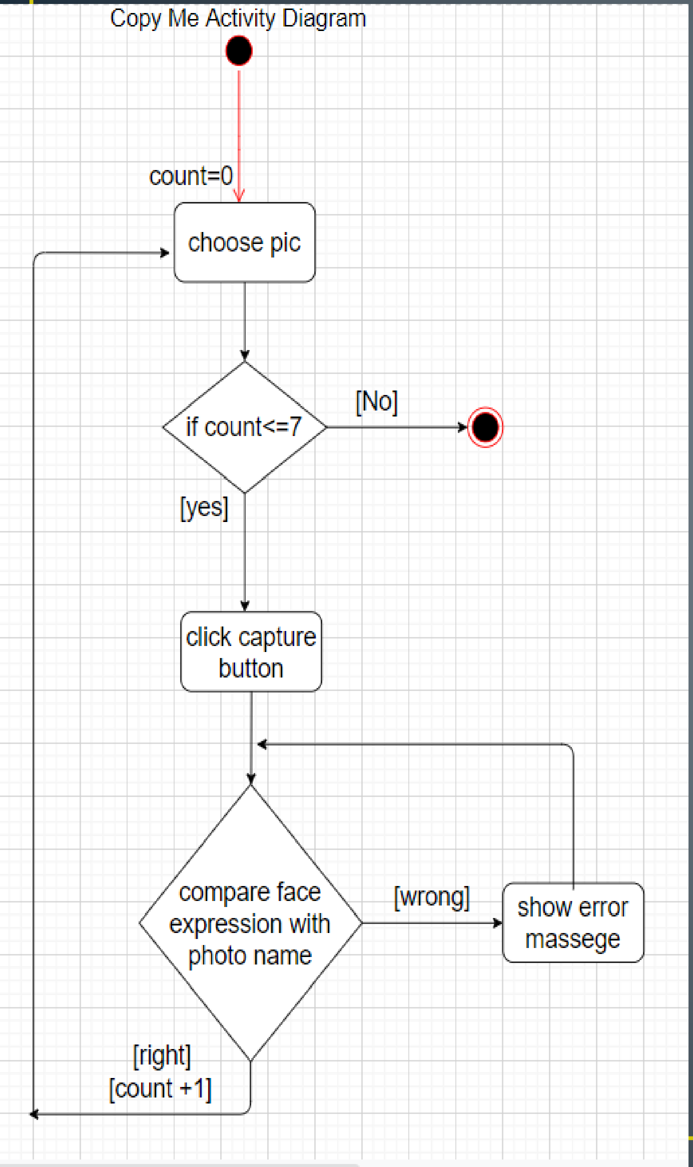
We used VGG model with Keras using TensorFlow backend

Also used batch size 32 and epochs 25

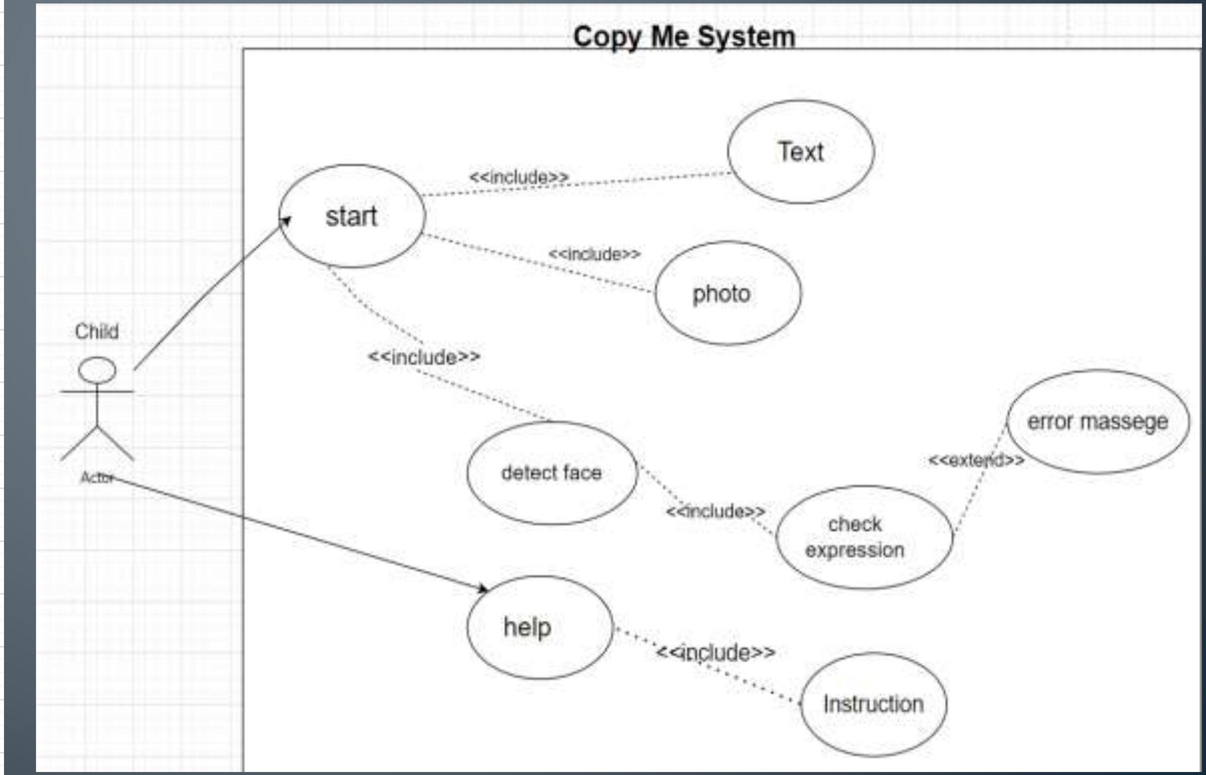


UML Diagrams:

Activity Diagram



Use Case Diagram



The Technologies:

We used:

- web cam in the desktop application and mobile cam in the mobile application
- We used Flutter in the mobile app and Tkinter in the desktop app
- Tensorflow, pillow, numpy, keras libraries
- Little VGG model to recognize facial expression

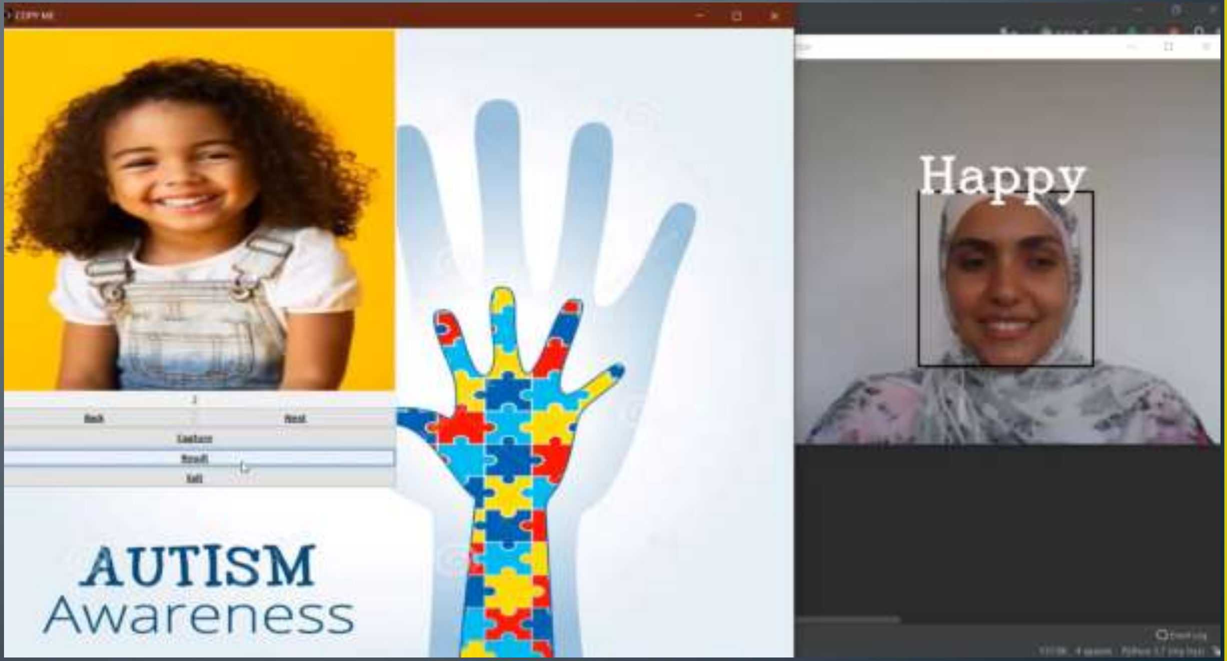
User Interface of Desktop Application:

The window after :

1-choosing a photo

2-clicking on (Capture) to turn on web cam and start capturing

3-clicking on (result) to get the user score



Example code for explanation:

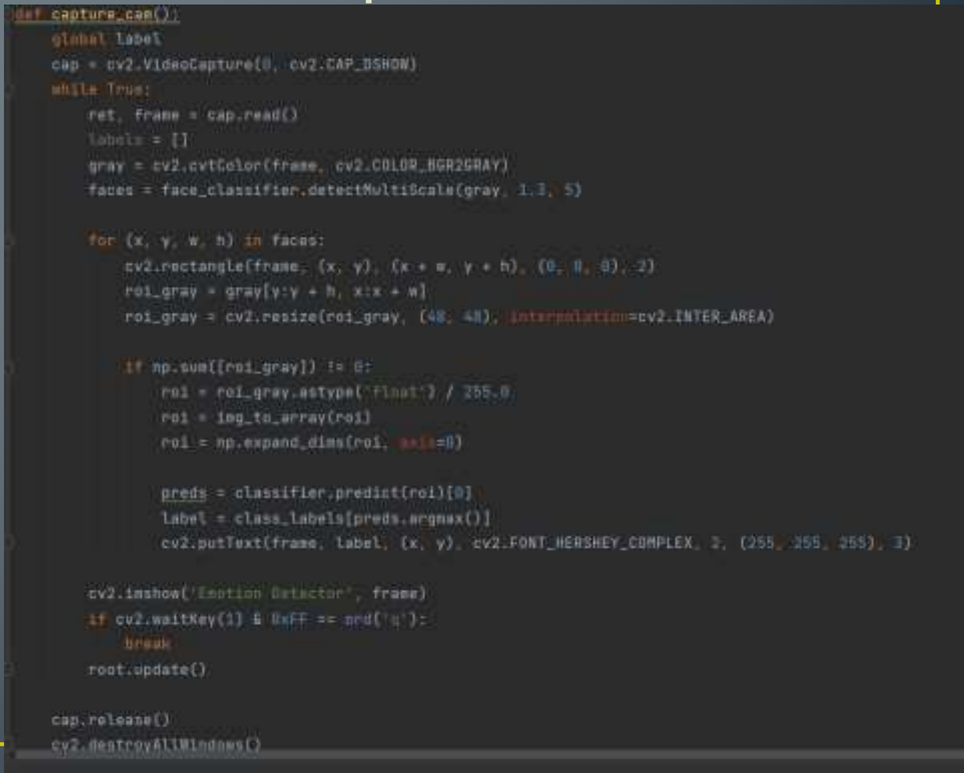
- In Desktop App:

1-In function (capture_cam):The user will choose when to open his web cam to start capturing his facial expression and retriving (label) which refers to the expression he really make

After that the user click

the button (result) to call

the function (get_score)



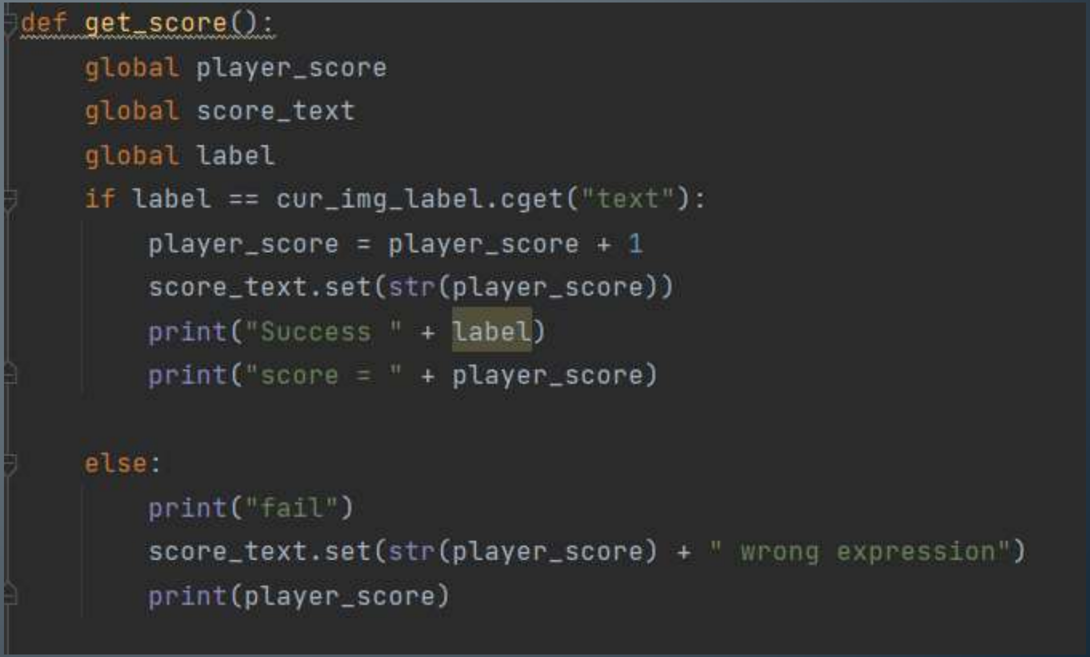
```
def capture_cam():
    global label
    cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
    while True:
        ret, frame = cap.read()
        labels = []
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_classifier.detectMultiScale(gray, 1.3, 5)

        for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 0), 2)
            roi_gray = gray[y:y + h, x:x + w]
            roi_gray = cv2.resize(roi_gray, (48, 48), interpolation=cv2.INTER_AREA)

            if np.sum([roi_gray]) != 0:
                roi = roi_gray.astype('float') / 255.0
                roi = img_to_array(roi)
                roi = np.expand_dims(roi, axis=0)

                preds = classifier.predict(roi)[0]
                label = class_labels[preds.argmax()]
                cv2.putText(frame, label, (x, y), cv2.FONT_HERSHEY_COMPLEX, 2, (255, 255, 255), 3)

        cv2.imshow('Emotion Detector', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
        root.update()

    cap.release()
    cv2.destroyAllWindows()
```

2- In function (get_score):

After clicking button(result), we comparing the “label” from model with “cur_img_label” name to see if the user made the right expression

-If (right) we increase the user score

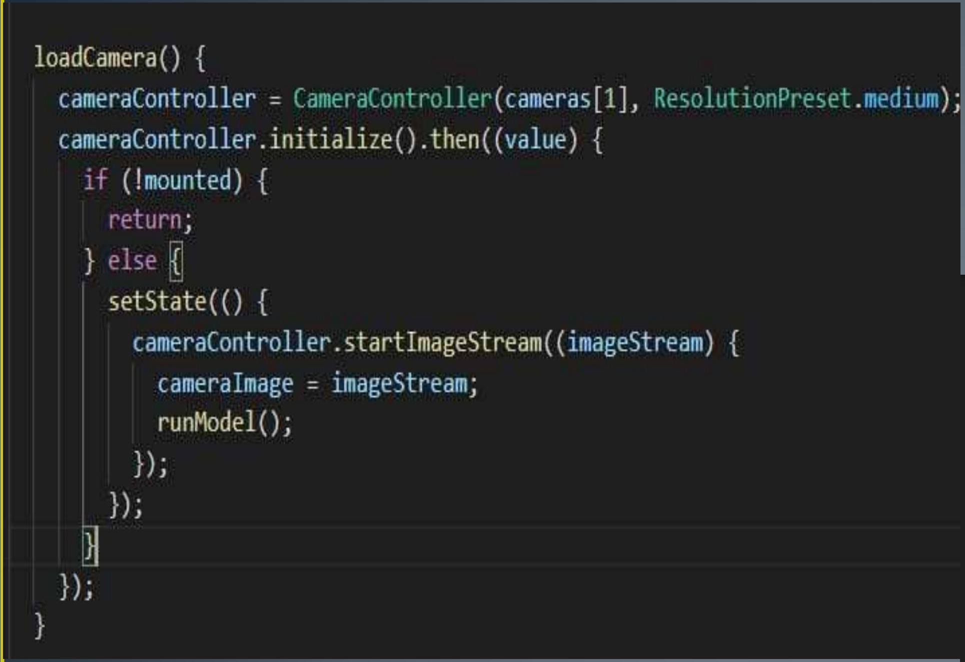
-If(wrong) we show an error message



```
def get_score():
    global player_score
    global score_text
    global label
    if label == cur_img_label.cget("text"):
        player_score = player_score + 1
        score_text.set(str(player_score))
        print("Success " + label)
        print("score = " + player_score)

    else:
        print("fail")
        score_text.set(str(player_score) + " wrong expression")
        print(player_score)
```

Mobile code explanation:

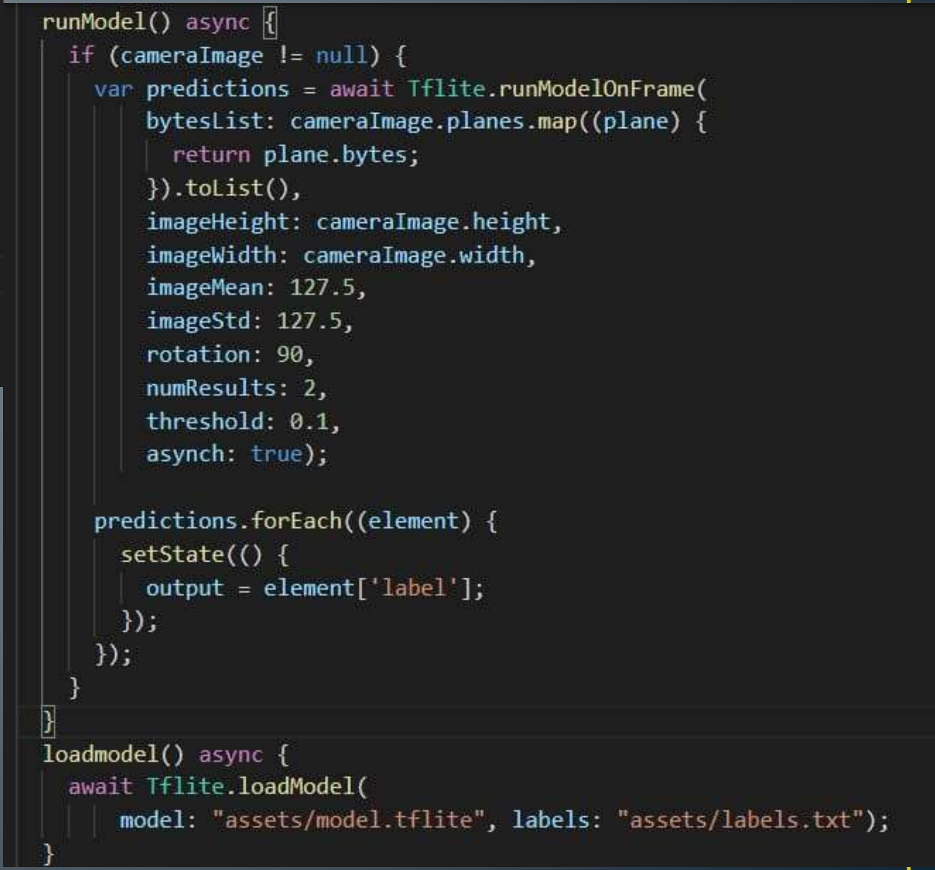
Most important functions



```
loadCamera() {
  cameraController = CameraController(cameras[1], ResolutionPreset.medium);
  cameraController.initialize().then((value) {
    if (!mounted) {
      return;
    } else {
      setState(() {
        cameraController.startImageStream((imageStream) {
          cameraImage = imageStream;
          runModel();
        });
      });
    }
  });
}
```

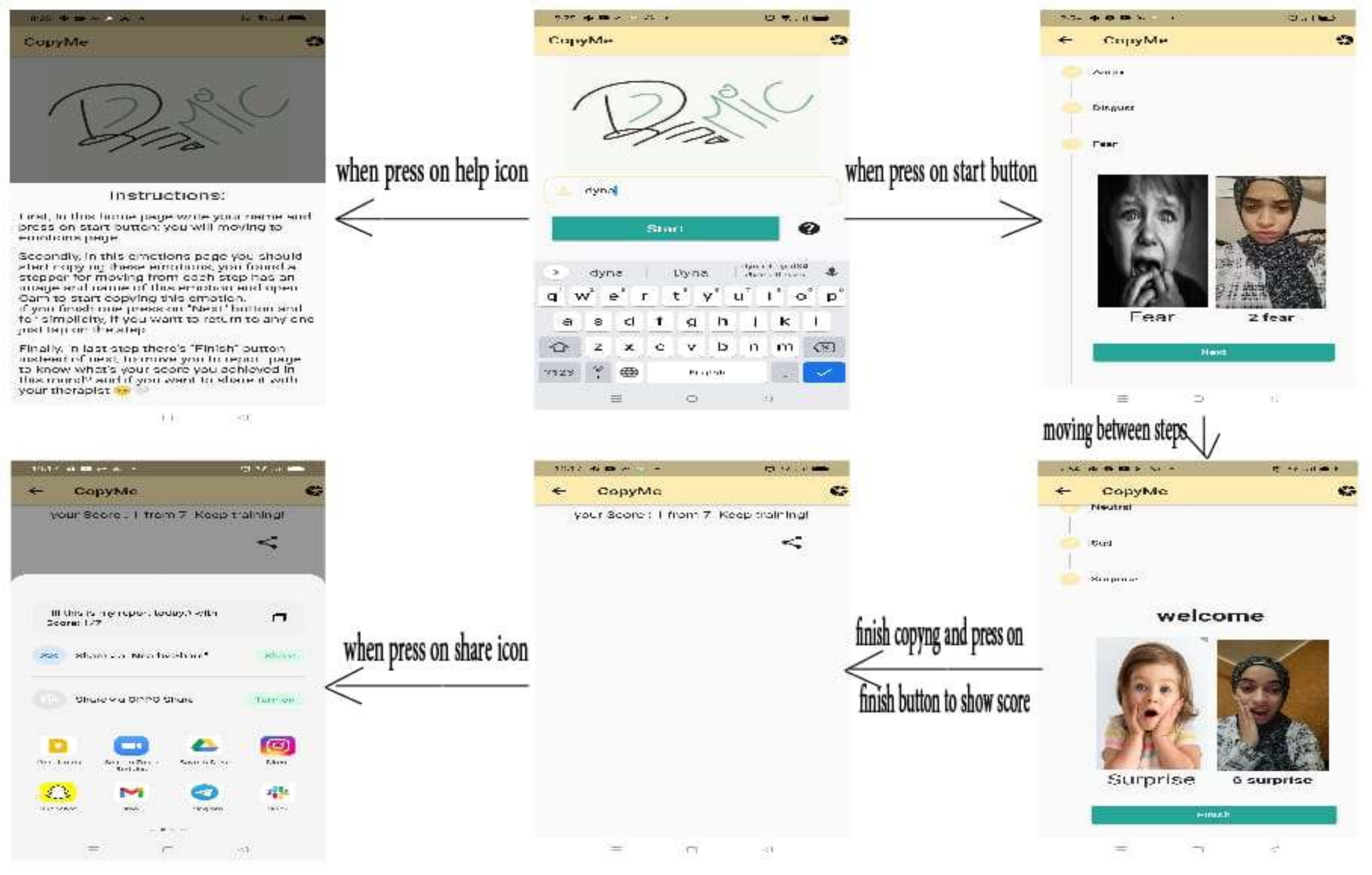
(Loadcamera) is for opening front mobile cam.

And calling (runModel) function



```
runModel() async {
  if (cameraImage != null) {
    var predictions = await Tflite.runModelOnFrame(
        bytesList: cameraImage.planes.map((plane) {
          return plane.bytes;
        }).toList(),
        imageHeight: cameraImage.height,
        imageWidth: cameraImage.width,
        imageMean: 127.5,
        imageStd: 127.5,
        rotation: 90,
        numResults: 2,
        threshold: 0.1,
        asynch: true);

    predictions.forEach((element) {
      setState(() {
        output = element['label'];
      });
    });
  }
}
loadmodel() async {
  await Tflite.loadModel(
      model: "assets/model.tflite", labels: "assets/labels.txt");
}
```

Wire Frame of Mobile Application:



Instructions:

First, in this home page write your name and press on start button: you will moving to emotions page

Secondly, In this emotions page you should start copying these emotions, you found a stepper for moving from each step has an image and name of this emotion and open Cam to start copying this emotion.
If you finish one press on "Next" button and for simplicity, If you want to return to any one just tap on the step

Finally, In last step there's "Finish" button instead of next, to move you to report page to know what's your score you achieved in this round!! and if you want to share it with your therapist

when press on help icon

when press on start button

moving between steps

finish copyng and press on finish button to show score

when press on share icon

your Score : 1 from 7 Keep training!

Hi this is my report today with Score: 1/7

6)Summary:

We are helping the Autistic children to recognize facial expression and be able to express them well

Future work:

We will make a database for each child record to let the therapist compare each child progress and let them know how to improve the child skills

References:

- https://www.researchgate.net/publication/260480073_CopyMe_an_Emotional_Development_Game_for_Children
- https://drive.google.com/folderview?id=1A8NkdJRZ6NCPg4dc01Kuzl_jkazOx8gc

Simulation of Autistic child amused by using Copy Me App