Real-Time Facial Emotion Recognition Using AI

Dana Nedal Ghazal
Computer Engineering Department
The University of Jordan
Amman, Jordan
DAN0183507@ju.edu.jo

Rawan Rami Hamdan
Computer Engineering Department
The University of Jordan
Amman, Jordan
ROA0183609@ju.edu.jo

Prof. Gheith Abandah
Computer Engineering Department
The University of Jordan
Amman, Jordan
abandah@ju.edu.jo

***Abstract*— Facial emotion recognition (FER) is a technology that analyzes facial expressions from images or videos to identify an individual's emotional state. One of the most important challenges faced by current artificial intelligence (AI) based mobile applications in recognizing facial emotions is bias as there is a lack of representation of faces wearing hijabs and Middle Eastern faces in the databases of faces used in available applications, which leads to inaccurate results of these applications. The proposed project aims to build a real-time emotion analysis application focusing on Middle Eastern faces particularly women wearing hijabs to increase its usability in the Arab world.**

The traditional method of emotion recognition may not meet the need of mobile application users for their emerging value-added services, so providing techniques for classifying facial expressions by taking advantage of the benefits of deep learning (DL) is important. We used a convolutional neural network (CNN) based model trained on the FER2013 dataset and retrained it on a hybrid dataset that includes the Extended Cohn-Kanade (CK+) dataset, Japanese female facial expression (JAFFE) dataset, and Iranian emotional face database (IEFDB). We achieved a test accuracy of 88% on the hybrid test set and 90% test accuracy on the IEFDB test set.

I. Introduction

The rapid growth of artificial intelligence (AI) has contributed a lot to the world of technology. Since traditional algorithms fail to meet human needs in real-time, machine learning (ML) and deep learning (DL) algorithms have gained great success in many applications like classification systems, recommendation systems, pattern recognition, *etc.* [1].

Facial emotion recognition (FER) is a technology that analyzes facial expressions from images or videos in order to reveal an individual's emotional state. Emotion plays a vital role in determining a person's thoughts, behavior, and feeling. In order to maximize user satisfaction, recognizing user emotion is the most important. Recognizing emotions has become an essential part of providing emotional care for people. Providing emotional care can greatly improve users' experience to improve quality of life. The traditional method of emotion recognition may not meet the need of mobile application users for their emerging value-added services, so providing techniques for classifying facial expressions by taking advantage of the benefits of DL is important [2, 3, 4].

Automatic analysis of facial expressions has attracted the interest of many AI researchers because these systems will have many uses in the fields of healthcare, gaming, and human-computer interaction (HCI) [5]. But this type of application has spread quite a bit in the Middle East and this is what led us to the idea of our project, where we wanted to build a real-time emotion analysis application focusing on Middle Eastern faces where women wear hijab to increase its usability in the Arab world, especially in Jordan.

II. Datasets

FER is a well-studied field, and there are several popular datasets available on the web that one can use to learn about facial expressions. Table 1 shows a comparison of the facial emotion recognition datasets we used in this project that include FER2013 dataset, CK+ dataset, JAFFE dataset, IEFDB dataset, and hybrid dataset depending on what includes, total number of images, number of classes, resolution, emotion classes, and source. The image representation of the datasets is shown in Fig. 1 such that each row represents an individual dataset and each column represents a different emotion [6, 7, 8].

TABLE 1. COMPARISON BETWEEN THE FACIAL EMOTION RECOGNITION DATASETS USED IN THIS PROJECT

Dataset	Includes	Total Number of Images	Number of Classes	Resolution	Emotion Classes	Source
FER2013	–	35,887	7	48x48	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Kaggle
CK+	–	902	7	640x490 or 640x480	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	University of California, Berkeley
JAFFE	–	213	7	256x256	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Kyushu University, Japan
IEFDB	–	248	7	5184x3456	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Tehran University of Medical Sciences, Iran
Hybrid	CK+, JAFFE, and IEFDB Datasets	1363	7	48x48	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	We Created It

Fig. 1. Sample Images for Each Emotion Class in the FER2013, CK+, JAFFE, and IEFDB Datasets

III. Datasets Preprocessing

Initially, we tried to understand how to preprocess the images of IEFDB to be applied to the ML model, to improve model performance and reduce the risk of overfitting. Therefore, we preprocessed the dataset images into the FER2013 dataset images format by resized the images to a size of 48x48 pixels, converted the color space of the images to grayscale, and normalized the pixel values in the images to a range between 0 and 1 by dividing each value by 255, then we split it into 80% for the training set and 20% for the test

set, then we split again the train set into 80% for the training set and 20% for the validation set. The following Table 2 shows the IEFDB splitting of the train, test, and validation dataset, depending on the split value and the number of images.

TABLE 2. THE SPLIT VALUE AND NUMBER OF IMAGES FOR TRAIN, TEST, AND VALID SET OF IEFDB

Dataset	Split Value	Number of Images
Train	60%	158
Test	20%	50
Validation	20%	40

Then to obtain a good accuracy score and to improve the performance and generalization ability of ML model, and reduce the impact of overfitting, also to increase diversity by covering a broader range of scenarios and conditions, we tried to increase the size of the training dataset by adding the CK+ dataset, and the JAFFE dataset. The CK+ dataset contains 902 face images of female and male between the ages of 18 and 50 with a variety of heritages, and have seven categories. As for the JAFFE dataset contains 213 images of Japanese female faces and seven categories, and we download them from GitHub website [9].

We started by preprocessing the images of these datasets into the FER2013 dataset images format by resizing them to 48x48 pixels, converted the color space of the images to grayscale, and normalized the pixel values in the images to a range between 0 and 1. Then we added them to the IEFDB, thus we have a hybrid dataset containing 1363 images from IEFDB, JAFFE, and CK+ datasets. We split it into 80% for the training set and 20% for the test set, then we split again the train set into 80% for the training set and 20% for the validation set. The following Table 3 shows the splitting of the hybrid dataset for the train, test, and validation dataset, depending on the split value and the number of images.

TABLE 3. THE SPLIT VALUE AND NUMBER OF IMAGES FOR TRAIN, TEST, AND VALID SET OF HYBRID DATASET

Dataset	Split Value	Number of Images
Train	60%	872
Test	20%	273
Validation	20%	218

IV. RETRAINING AND TESTING THE CNN MODEL

Using the Python programming language, we retrain and test the ML model with different sizes of the training set and test set. The package used for split was imported from the Scikit-learn library. Using the TensorFlow library, we loaded a pre-trained kaggle_model that we found on Kaggle website [10] and was trained on the FER2013 train set, then we started retraining this model on two different sizes of training sets. The accuracy of the model was calculated and analyzed for each training set.

V. MOBILE APPLICATION DEVELOPMENT

We developed the facial emotion recognition mobile application named Emotional ID using Android Studio, chose Java programming languages to develop the application, and used extensible markup language (XML) code to design the user interface (UI).

A. App Design and How it Works

Our Android app is simple enough to use as it will consist of five screens. In the first screen, the user will see the name of the app and the logo as shown in Fig. 2(a), then it will move to the second screen that contains two buttons, the first to take the photo with the camera, and the other to detect the user's emotions in real-time as shown in Fig. 2(b).

When press the first button, a screen will appear with three buttons, one to capture an image, the other to choose an image from gallery, and the last to process the image as shown in Fig. 2(c). Once the image has been processed, you will proceed to the next screen, which displays the image captured, the user's emotion, and tips on how to act positively while experiencing that feeling. In addition, there is a button that, when clicked, will open a video for the user to watch to help him improve his emotional intelligence and motivate him to improve his mood.

When the second button is pressed, a camera screen will appear for the user, which will detect his emotions in real-time. At the top of the camera there will be a button to flip the camera and next to it there is a button to set the delay time, and based on the time that the user sets, a notification will be sent to the user containing the user's emotion and some tips, and when clicking on the notification, a video will open for the user to watch to give him a visual reference to understand how to manage his feeling.

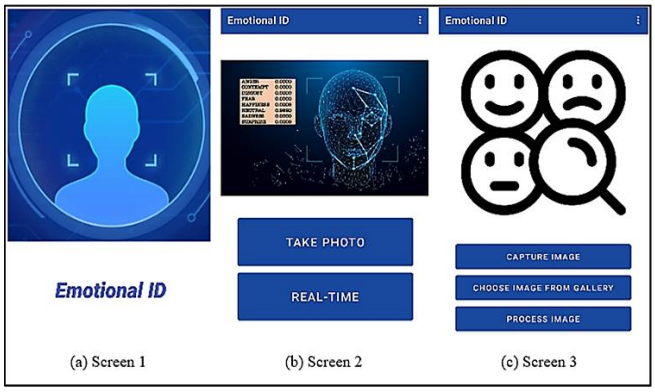


Fig. 2. The Design of the Application Screens

B. Convert Keras Model to TensorFlow Lite

To integrate our Keras model into our Android app, we used TensorFlow Lite (TFLite) which can be a powerful image recognition and analysis tool. By leveraging the power of ML on mobile devices, developers can create powerful and innovative applications that can help users solve complex problems and make better decisions [11].

TensorFlow Lite is a lightweight version of the popular TensorFlow library designed specifically for mobile and embedded devices. By converting a pre-trained Keras model from .h5 file extension into .tflite file extension, developers can run the model on mobile devices in real-time without requiring internet connectivity or a remote server [12, 13].

To convert a Keras model to TFLite format for use in an Android application, we used the TFLiteConverter module provided by TensorFlow. The TFLiteConverter module provides a simple and straightforward API for converting Keras models to the TFLite format. First, we install the TensorFlow Lite converter and import it into our Python code. Next, we define the model that needs to be converted to TFLite format, and then use the TFLiteConverter module to convert the Keras model to TFLite format [12, 13]. Once the model is in the TensorFlow Lite format, it can be added to the Android application as a model file. The application can then use the TensorFlow Lite interpreter to load the model and make predictions on new data. The TensorFlow Lite interpreter can then be used to run inference on the model and generate predictions in real-time [12].

C. *Face Detection*

We used the haarcascade_frontalface_alt.xml file available on GitHub website [14] in our application for face detection, which is a pre-trained model classifier trained on a large dataset of labeled face images for detecting frontal faces in images and videos. The model uses adaptive boosting (AdaBoost) algorithm in order to yield better results and accuracy. It is part of the open-source computer vision (OpenCV) library and ML library. This classifier can be used in various computer vision applications, such as security systems, facial recognition, augmented reality, and facial expression recognition. The file can be used in various programming languages that support OpenCV, including Python, and Java [15].

To use this file for face detection in our app, we followed several steps. First, we download the haarcascade_frontalface_alt.xml file from GitHub and include it in the Android project directory. Next, we initialize the cascade classifier object in the application code and load the haarcascade_frontalface_alt.xml file into it. Next, we read the image in which faces will be detected and converted it to grayscale. Finally, we used the DetectMultiScale method of the cascading classifier object to detect faces in grayscale images. The DetectMultiScale method returns the position and size of the detected object as a bounding box or a set of rectangles. These rectangles represent the regions in the image where the object is likely to be present [16].

Fig. 3 illustrates the process that the image goes through. First, the image is fed into the haar cascade model, which detects the face in the image, then fed into the CNN model, which predicts the emotion of the input image.

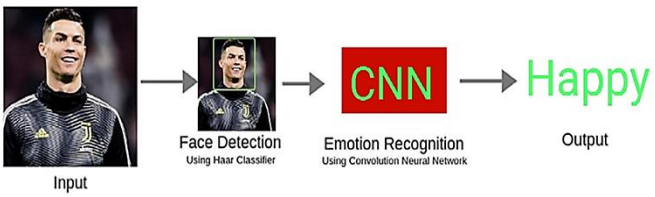


Fig. 3. The Steps an Image Goes Through to Detect Facial Emotions

VI. Model Results

A. *The Results of the Model Retrained on the IEFDB Dataset*

After we loaded the pre-trained kaggle_model, we selected the batch size 40 and epochs 400. To prevent overfitting and improve the generalization ability of the model during training, we used two callback functions: Early Stopping and ReduceLROnPlateau. These functions helped us determine the optimal stopping point, and adjust the learning rate when progress stagnated. For compiling we used Adam as the optimizer with a learning rate of 0.01, and the categorical cross-entropy as the loss function. The model hyperparameters are shown in Table 4.

TABLE 4. MODEL HYPERPARAMETERS RETRAINED ON THE IEFDB TRAIN SET

Model Hyperparameters	**Values**
Learning Rate	0.01
Number of Epochs	400
Batch Size	40
Optimizer	Adam
Loss Function	Categorical Cross-entropy

We first tested it on the test set of the IEFDB dataset and we achieved 28% test accuracy, also we tested it on the test set of the hybrid dataset and we achieved 16% test accuracy, then we retrained the model on the training set of the IEFDB dataset and evaluated the model, we achieved 92% accuracy on the IEFDB training set, 74% accuracy on the IEFDB test set, and 58% accuracy on the IEFDB valid set.

Also, after retraining the model, we retested it on the test set of the hybrid dataset and we achieved 33% test accuracy, in order to ensure the accuracy of the model's recognition of faces for all genders without exception, including images of women wearing the hijab. The briefly accuracy of the model on training, test, and validation sets show in Table 5.

TABLE 5. ACCURACIES OF THE MODEL RETRAINED ON THE IEFDB TRAIN SET

Dataset	**IEFDB Dataset Accuracy**	**Hybrid Dataset Accuracy**
Train	92%	–
Test Before Retraining the Model	28%	16%
Test After Retraining the Model	74%	33%
Validation	58%	–

Based on the results presented in the table, we note the improvement of the accuracy values of the model retrained on the IEFDB dataset, also we note the improvement in the test accuracy of the hybrid dataset that consisting of IEFDB, CK+, and JAFFE datasets after retraining the model compared to before.

Fig. 4 shows the accuracy and loss for the model that retrained on the training set of the IEFDB dataset the figure explains the increase in both the accuracy of training and validation as the model progressed through the training epochs, and the figure illustrates a decrease in loss for each of the training and validation, this indicates the model's ability to minimize prediction errors. The figure also highlights the effect of the Early Stopping callback function, which halted the training process at epoch 71.

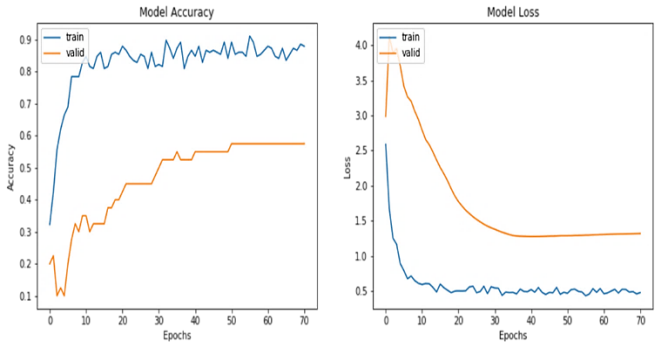


Fig. 4. Model Accuracy and Loss for Training and Validation Sets for IEFDB Dataset

Fig. 5 also displays the confusion matrix representing the number of TP, TN, FP, and FN, where the values on the diagonal represent the correct predictions made by the model. As we can see most of the false predictions are in surprise emotion where surprise is mostly predicted as neutral or sad. The second largest false predictions are in neutral emotion as neutral is sometimes predicted to be sad. We also note that the model did not predict any correct predictions of the emotion of fear, due to the small number of images of the fear emotion.

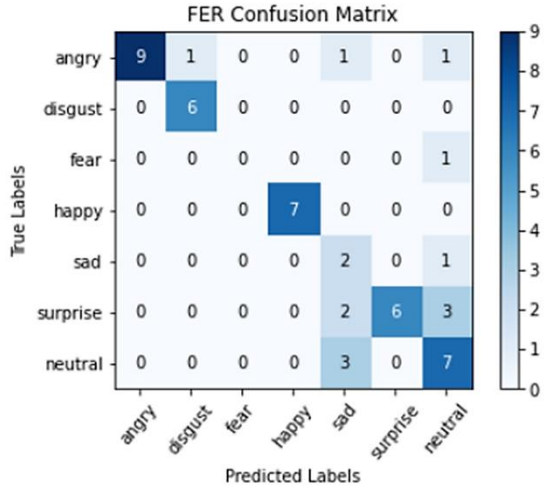


Fig. 5. Confusion Matrix of the Model for IEFDB Test Set

The classification accuracy and average results on the IEFDB test set for the model retrained on the IEFDB train set based on precision, recall, and F1 score are provided in Table 6. The table shows that emotions like angry, disgust, happy, and surprise have achieved high precision, recall, and F1 scores, indicating that the model accurately identified instances of these emotions. On the other hand, the model struggles with emotions like fear and sad, where the model's performance is considerably lower.

TABLE 6. CLASSIFICATION RESULTS OF THE MODEL ON THE IEFDB TEST SET

Emotion	Precision	Recall	F1 Score
Angry	100%	75%	86%
Disgust	86%	100%	92%
Fear	0%	0%	0%
Happy	100%	100%	100%
Sad	25%	67%	36%
Surprise	100%	55%	71%
Neutral	54%	70%	61%
Average	66%	67%	64%

B. The Results of the Model Retrained on the Hybrid Dataset

After we loaded the pre-trained kaggle_model, we compiled the model using Adam's optimizer with a learning rate of 0.001, and the categorical cross-entropy as the loss function. We chose the number of epochs 300 and batch size 70. To prevent overfitting and improve model performance, we used three callback functions: Early Stopping, ReduceLROnPlateau, and Model Checkpoint. These functions helped us determine the optimal stopping point, adjust the learning rate when progress stagnated, and save the best model checkpoint, respectively. Table 7 shows the model hyperparameters.

TABLE 7. MODEL HYPERPARAMETERS RETRAINED ON THE HYBRID TRAIN SET

Model Hyperparameters	Values
Learning Rate	0.001
Number of Epochs	300
Batch Size	70
Optimizer	Adam
Loss Function	Categorical Cross-entropy

Then we tested it on the test set of the hybrid dataset and achieved 19% test accuracy, and tested it on the test set of the IEFDB so we achieved 34% test accuracy, these result before training the kaggle_model. Then we retrained the model on the training set of the hybrid dataset and evaluated the model, we achieved 88% test accuracy on the hybrid test set, 99% train accuracy on the hybrid train set, and 87% valid accuracy on the hybrid valid set.

Also, after retraining the model, we retested it on the test set of the IEFDB and we achieved 90% test accuracy, in order to ensure the accuracy of the model's recognition of Middle Eastern faces, and the faces of women who wear the hijab. Table 8 demonstrates our accuracy gains and the comparison of train accuracy, validation accuracy, and test accuracy before and after training kaggle_model on the hybrid dataset. The results, show that the test accuracy after retraining the model is higher than the test accuracy before retraining the model, and the model demonstrated its ability to recognize emotions in Middle Eastern faces and women wearing the hijab, as indicated by the high-test accuracy on the IEFDB test set after retraining the model.

TABLE 8. ACCURACIES OF THE MODEL RETRAINED ON THE HYBRID TRAIN SET

Dataset	IEFDB Dataset Accuracy	Hybrid Dataset Accuracy
Train	–	99%
Test Before Retraining the Model	34%	19%
Test After Retraining the Model	90%	88%
Validation	–	87%

Fig. 6 shows the accuracy and loss for the model that retrained on the training set of the hybrid dataset the figure explains the increase in both the accuracy of training and validation as the model progressed through the training epochs, and also shows a decrease in loss for each of the training and validation, also the Early Stopping made it stop training at epoch 118.

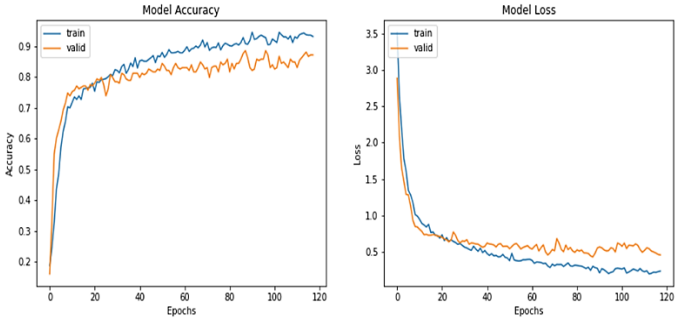


Fig. 6. Model Accuracy and Loss for Training and Validation Sets for the Hybrid Dataset

Also, the confusion matrix of the model is shown in Fig. 7 that displays the number of TP, FP, TN, and FN for each class. As we can see most of the false predictions are in neutral emotion where neutral is mostly predicted as sad or disgust.

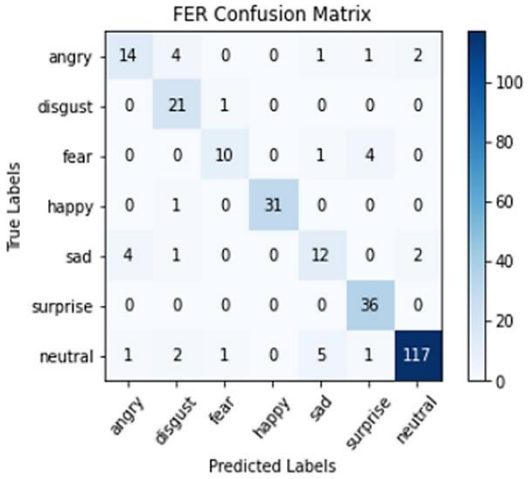


Fig. 7. Confusion Matrix of the Model for Hybrid Test Set

The classification accuracy and average results on the hybrid test set for the model retrained on the hybrid train set based on precision, recall, and F1 score are provided in Table 9. The table indicates that the model performs well in detecting happy and neutral emotions. However, it struggles a bit to detect emotions like angry and sad.

TABLE 9. CLASSIFICATION RESULTS OF THE MODEL ON THE HYBRID TEST SET

Emotion	Precision	Recall	F1 Score
Angry	74%	64%	68%
Disgust	72%	95%	82%
Fear	83%	67%	74%
Happy	100%	97%	98%
Sad	63%	63%	63%
Surprise	86%	100%	92%
Neutral	97%	92%	94%
Average	82%	83%	82%

C. Discussion the Results

Be the most important results that we reached based on the accuracy values that we obtained. On the basis of the above graphs and tables, which showed the improvement in model predictions after increasing the size of the dataset and adding two JAFFE and CK+ datasets with the IEFDB dataset, and we called it a hybrid dataset, then we divided it into a training and test sets, in order to evaluate the ML model performance on the independent test set that it has not seen before. We also divided the training set into training and validation sets for helps to ensure that the model is not overfitting to the training data and that it will perform well on new, unseen data.

The first model, retrained only on the IEFDB dataset with a learning rate of 0.01 and batch size of 40 and epochs 400, may converge faster and overfit more easily to the specific features and patterns of the IEFDB dataset. However, it may not generalize to new datasets or different images of the same emotion due to the limited training data and hyperparameters.

The second model, retrained on the hybrid dataset which is larger and more diverse dataset consisting of the IEFDB, CK+, and JAFFE datasets with a smaller learning rate of 0.001 and a larger batch size of 70 and epochs 300, may take longer to converge but may also lead to better generalization to new datasets. The larger dataset helps the model learn more robust and representative features and patterns of different emotions, while the smaller learning rate and larger batch size can help prevent overfitting and improve the model's stability.

In summary, while both models may have different strengths and weaknesses depending on the specific datasets and hyperparameters, the second model retrained on a larger and more diverse dataset with a smaller learning rate and larger batch size have better generalization and performance on new datasets and images of different emotions.

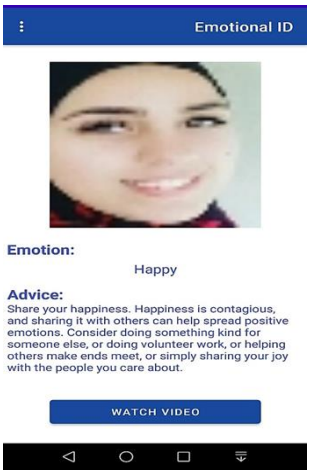
Finally, after reviewing the accuracy results that came out with us, which for the first model is 74% test accuracy on the IEFDB test set, 33% test accuracy on the hybrid test set. For the second model is 90% test accuracy on the IEFDB test set, and a test accuracy of 88% on the hybrid test set. We therefore decided to use the second model that was retrained on the hybrid dataset, containing IEFDB, JAFFE, and CK+ datasets in our app because it showed higher accuracy, better performance, and less overfitting than the first model that retrained on the IEFDB dataset. The retrained model on the hybrid dataset can also recognize the feelings of most faces of different genders and races, including the faces of women wearing hijab and Middle Eastern faces that our project is focusing on because it was retrained on a large dataset that contained diversity in face images unlike the model retrained on IEFDB only.

Comparing the result that we reached with the previous works, we find that the highest accuracy that was reached was for the previous work CS230-FER [17]. In which they trained the model on FER2013, CK+, and JAFFE datasets and achieved a test accuracy of 75.8%, while we used the pre-trained kaggle_model on the FER2013 dataset and its test accuracy is 67.05%, and we retrained it on CK+, IEFDB and JAFFE datasets, and we achieved 88% test accuracy, and thus we obtained a higher accuracy than the previous works, which is the highest to our knowledge.

VII. APPLICATION RESULTS

Our Emotional ID mobile application identifying the user's feelings can provide valuable feedback on the effectiveness and accuracy of the technology. In the first scenario, where the user can take a photo from the camera or choose photo from gallery, then the app will show the image, displaying the detected emotion below it, while showing advice for the user on how to act positively while experiencing that feeling. In addition, at the bottom of the screen there is a button that when the user clicks on it opens a video to help the user improve their emotional intelligence and mood. Fig. 8(a) shows the result of the sad face image that taken by camera, also Fig. 8(b) shows the result of the happy face image that chosen from gallery, and how the advice and the watch video button appear in the screen.

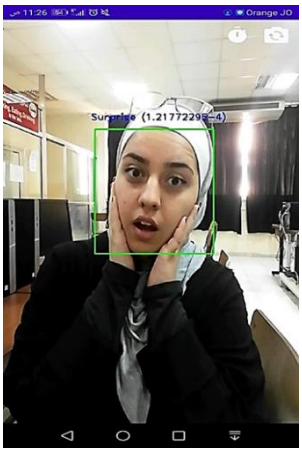
(a) Image That Take by Camera



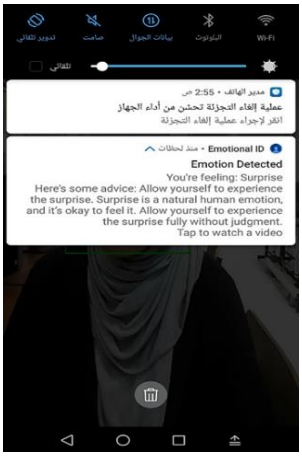
(b) Image That Choose from Gallery

Fig. 8. The Results of Emotion Recognition in Two Cases of Take Photo

In the second scenario, where the app detects the user's emotions in real-time, a camera screen appears, with the flip camera button and delay time to send notification setting option located at the top, and based on the delay time set by the user, the app sends a notification containing the user's emotion and tips for managing it, which can be clicked to watch a helpful video. Fig. 9(a) and Fig. 9(c) show the result for the real-time emotion detection also Fig. 9(b) and Fig. 9(d) show how the notification appears on the screen.

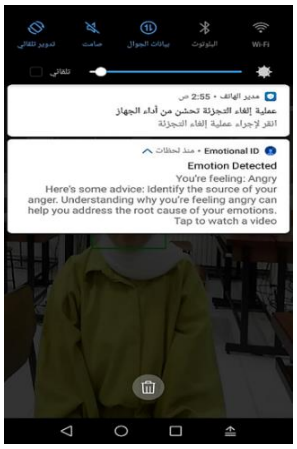


(a)Surprise Emotion in Real-Time



(b) Surprise Notification

(c)Angry Emotion in Real-Time

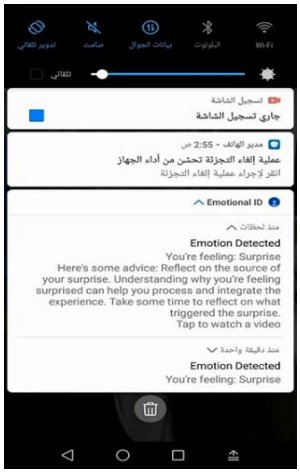


(d) Angry Notification

Fig. 9. The Results of Real-Time Emotion Recognition

Fig. 10(a) show the times to delay sending the notification so that the user can choose one of these times. Here we chose one minute, and Fig. 10(b) shows that the notification was sent after one minute.

(a) Choose a Delay of One Minute



(b) The Notification Appears

Fig. 10. Select Delay Time and Send Notification

However, there are many issues our app may have in recognizing emotions as it is not 100% accurate. One major problem is that emotions are complex and subtle, and people express them differently, which can lead to difficulties in accurately interpreting and predicting emotions. In addition, the accuracy of the emotion detection feature in the application can also be affected by factors such as lighting, camera quality, image quality, and the user's facial expressions, which must be taken into account when evaluating the results of the emotion detection feature in the application, which can lead to incorrect results.

Our application is distinguished from previous applications in that it takes cultural differences into account so that there is no bias in the training data used. Our application can accurately recognize the feelings of East Asian faces and faces that have European features, and focus on Middle Eastern faces and faces that wear the hijab, which are faces that do not other applications recognize her feelings correctly. Where our application is characterized by providing advice and giving useful videos to improve the user's feeling, as in the case of detecting emotions in real-time, it sends a notification that contains the user's feeling and gives him the appropriate advice, and these features are absent in previous works.

VIII. Conclusions

In conclusion, our project addressed the challenge of identifying facial expressions by developing a real-time facial emotion recognition application that is focused on Middle Eastern faces, especially for women wearing hijab, to overcome the bias and inaccuracy issues that current AI-based applications have. The project's importance lies in the fact that facial expressions are a key part of nonverbal communication, and the inability to perceive these expressions leads to a loss of communication effectiveness. Our project utilized the efficient CNN algorithm to analyze facial expressions and identify the emotional state of the person in real-time. This system has potential applications in various fields, including healthcare, gaming, and marketing. Additionally, it can also be useful in online education, where capturing students' affect can contribute to the effectiveness of teaching. By recognizing facial emotions, we can gain insight into the minds of people we interact with, even when they are not talking, and this information is important for many professionals.

We explored and retrained the model on a different dataset, and it turned out that the retrained model on a large dataset containing a variety of facial images is the most accurate, as the results of the project showed that the model that was retrained on the hybrid dataset achieved a test accuracy of 88% on the hybrid test set and 90% test accuracy on the IEFDB test set, which is the highest based on the research we have read. In addition, we demonstrated that FER models can be applied in the Arab world by developing an Android application with real-time recognition speeds that recognizes the emotions of different faces including Middle Eastern faces and the faces of women wearing hijab. The successful implementation of our project provides a foundation for future research in this field, and we hope that our work will be beneficial for both academic and industrial purposes.

IX. Future Work

Our project has focused on developing a real-time emotion analysis application focusing on Middle Eastern faces where women wear the hijab to increase its usability in the Arab world. However, there are several ways in which our solution can be further improved to enhance its accuracy, usability, and effectiveness. One direction for future work is to expand the dataset used for training the DL model. Although we collected a large number of images for our dataset, it is still limited in terms of diversity.

Another direction for future work is to develop a system for continuous monitoring of emotions. Our current application provides real-time emotion analysis, but it is not linked to cloud services to store the user's emotions and keep track of their mood flow weekly or monthly and give them an arithmetic average of the emotions they felt over that time period. Furthermore, the application can be integrated with wearable devices, such as smartwatches or fitness trackers, to provide continuous monitoring of emotions throughout the day, such as monitoring the emotions of students during an online class or detecting the emotional state of patients in a hospital.

Finally, integration of acoustic information since emotions can be conveyed through voice, so we can improve the accuracy of our emotion recognition system by integrating vocal information with facial expression recognition. This can be done by using verbal emotion recognition techniques that can help provide additional information to the model. Body language can also be used to convey feelings and combine them with facial expressions. Thus, the application can be developed to make it discover the user's feeling from his voice or from his body language. We believe that implementing these suggestions will make a significant contribution to the field of facial emotion recognition, especially in the Middle Eastern region.

X. References

[1] E. Pranav, S. Kamal, C. S. Chandran, and M. H. Supriya, "Facial emotion recognition using deep convolutional neural network," In 2020 6th International conference on advanced computing and communication Systems (ICACCS), pp. 317-320, IEEE, 2020.

[2] EDPS, K. Vemou, and A. Horvath, "Facial Emotion Recognition," 2021. [Online]. Available: https://edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-12021-facial-emotion-recognition_en. [Accessed 11 November 2022].

[3] I. K. Choi, H. E. Ahn, and J. Yoo, "Facial expression classification using deep convolutional neural network," Journal of Electrical Engineering and Technology, 13(1), pp. 485-492, 2018.

[4] M. Shamim Hossain, and G. Muhammad, "An emotion recognition system for mobile applications," IEEE Access 5, pp. 2281-2287, 2017.

[5] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," In 2005 IEEE international conference on multimedia and Expo, pp. 5-pp, IEEE, 2005.

[6] F. Heydari, S. Sheybani, and A. Yoonessi, "Iranian Emotional Face Database: Acquisition and Validation of a Stimulus Set of Basic Facial Expressions," Behavior Research Methods, 2022.

[7] Kaggle, [Online]. Available: https://www.kaggle.com/datasets/msambare/fer2013. [Accessed 7 February 2023].

[8] I. Dagher, E. Dahdah, and M. Al Shakik, "Facial expression recognition using three-stage support vector machines," Visual Computing for Industry, Biomedicine, and Art 2(1), 1-9, 2019.

[9] GitHub, [Online]. Available: https://github.com/spenceryee/CS229. [Accessed 27 February 2023].

[10] Kaggle, [Online]. Available: https://www.kaggle.com/code/aroop123/face-emotion-recoginition/data. [Accessed 27 February 2023].

[11] Seeedstudio, "Getting Started with TensorFlow Lite on reTerminal," [Online]. Available: https://wiki.seeedstudio.com/reTerminal_ML_TFLite/. [Accessed 28 March 2023].

[12] Studymachinelearning, "Introduction to TensorFlow Lite," [Online]. Available: https://studymachinelearning.com/introduction-to-tensorflow-lite/. [Accessed 28 March 2023].

[13] Tensorflow, "Convert TensorFlow models," [Online]. Available: https://www.tensorflow.org/lite/models/convert/convert_models. [Accessed 28 March 2023].

[14] GitHub, [Online]. Available: https://github.com/opencv/opencv/tree/master/data/haarcascades. [Accessed 28 February 2023].

[15] Ersanpreet, S. Singh, "Face and eyes detection with Viola Jones along with python code," 10 March 2020. [Online]. Available: https://ersanpreet.wordpress.com/tag/haarcascade_frontalface_default-xml/. [Accessed 28 February 2023].

[16] Towardsdatascience, G. Behera, "Face Detection with Haar Cascade," 24 Dec 2020. [Online]. Available: https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08. [Accessed 28 February 2023].

[17] A. Khanzada, C. Bai, and F. Celepcikay, "Facial expression recognition with deep learning," arXiv preprint arXiv:2004.11823, 2020.