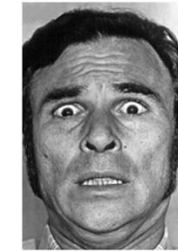


# PROJECT OVERVIEW

- In this project, we develop a deep learning system for classifying facial emotions using 7 basic emotions such as: (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral)
- This involves leveraging advanced machine learning algorithms and image processing techniques to accurately detect and interpret emotions from facial expressions.



Fearful



Angry



Sad



Happy



Disgusted



Surprised

## RESEARCH AND IMPACT

- This project helps us better understand human emotions, offering valuable insights for psychological research.
- It shows promise for mental health monitoring and therapy, providing tools to improve emotional well-being.
- Ultimately, it advances empathetic AI and robotics, leading to more human-like interactions with technology.

# DATASET

- For this project, we use the **FER 2013** dataset from Kaggle
- The data consists of **48x48** pixel grayscale images of faces.
- The faces have been automatically aligned so that each face is centered and occupies a similar amount of space in each image.
- The training set consists of **28,709** examples and the public test set consists of **3,589** examples.



kaggle

## FER2013 Dataset



Classes	Train	Test
Happy	7215	1774
Sad	4830	1247
Fear	4097	1024
Surprise	3171	831
Neutral	4965	1233
Angry	3995	958
Disgust	436	111

# EXPERIMENTIAL SETUP

## Data Reader

```
def read_images(self):
    print("Reading Data...")
    self.train_X, self.train_Y = self.load_data("data/train")
    self.test_X, self.test_Y = self.load_data("data/test")

    self.train_X = np.asarray(self.train_X) / 255.0
    self.train_Y = np.asarray(self.train_Y)
    self.test_X = np.asarray(self.test_X) / 255.0
    self.test_Y = np.asarray(self.test_Y)

    # One-hot encode the labels
    num_classes = len(self.label)
    self.train_Y = to_categorical(self.train_Y, num_classes)
    self.test_Y = to_categorical(self.test_Y, num_classes)
```

## Hyperparameter

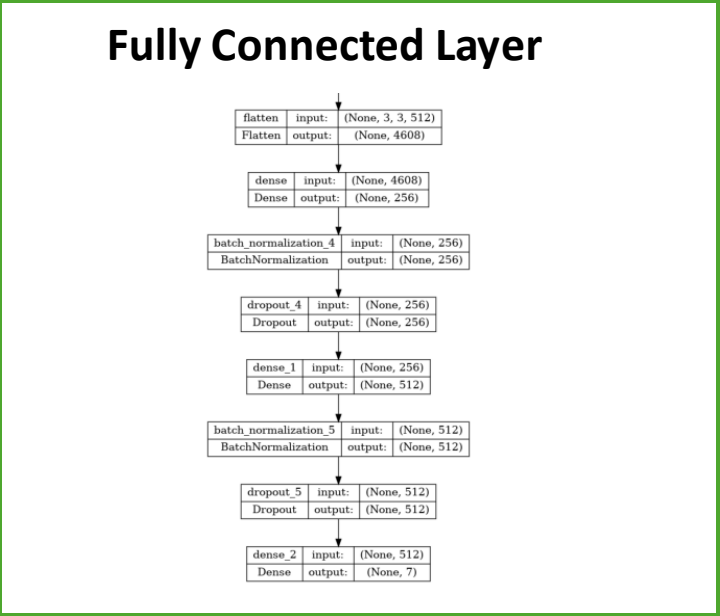
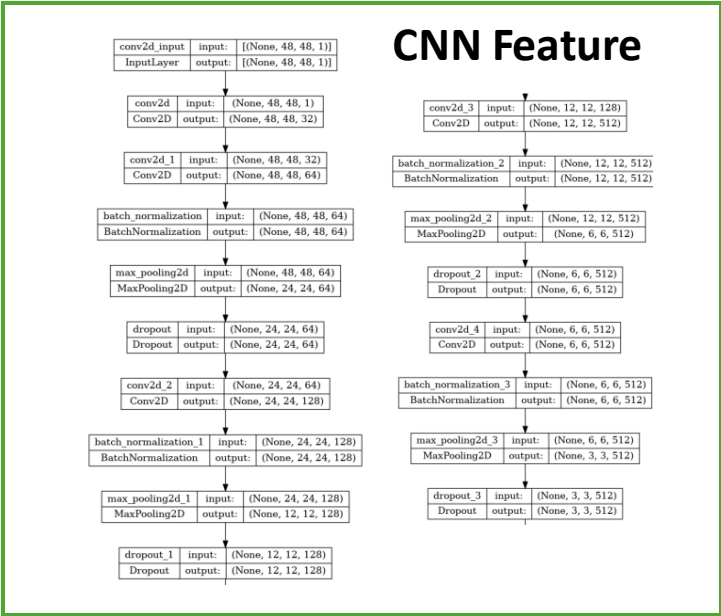
Epoch	65
Optimizer	Adam
Loss Function	Categorical Cross Entropy
Learning Rate	0.0001
Metrics	Accuracy

## Train Model

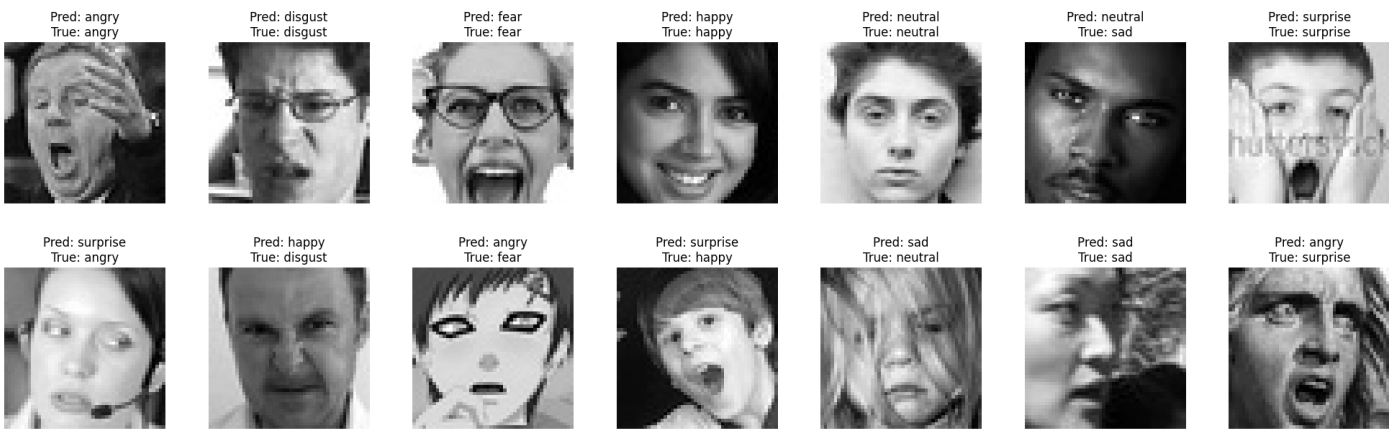
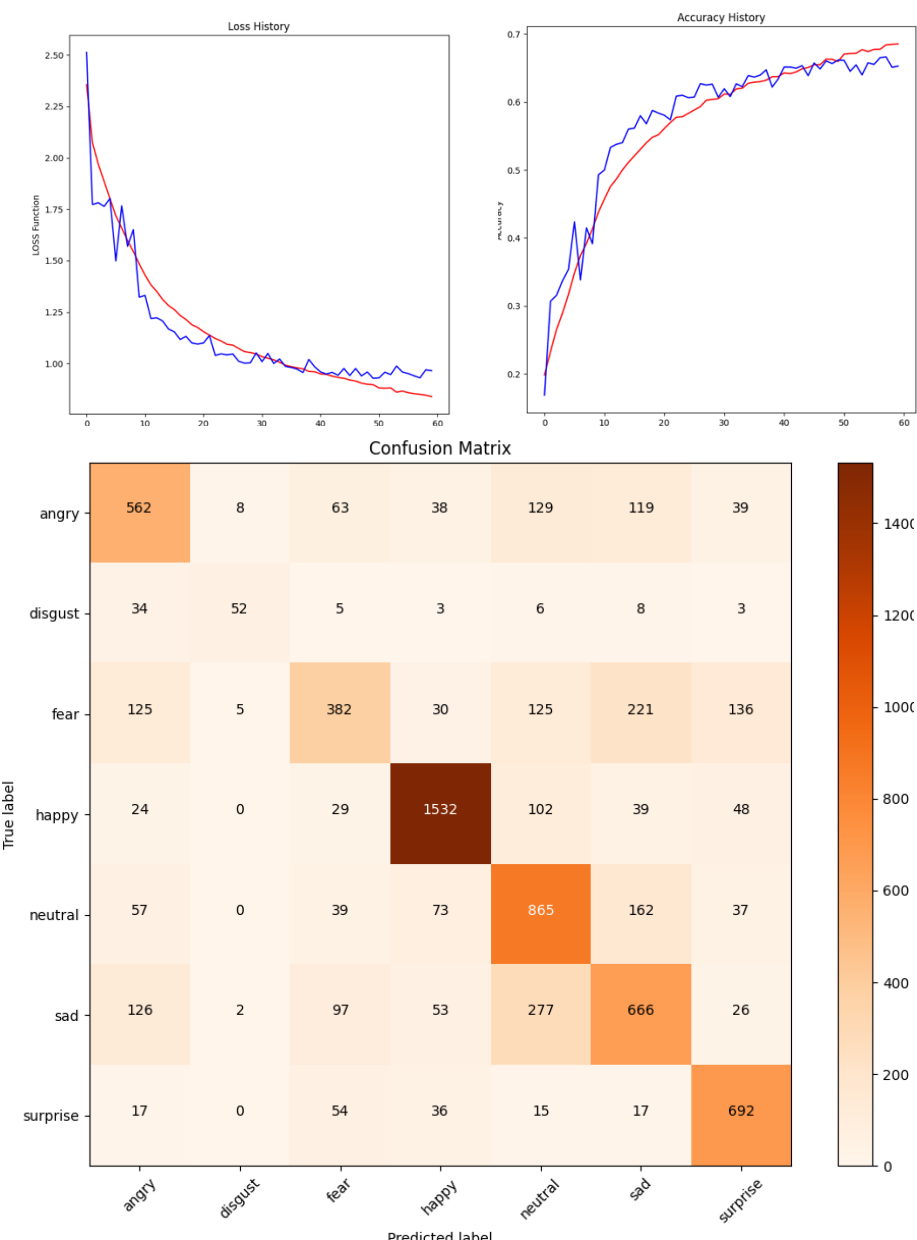
```
# Compile the neural network.
model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.0001),
              metrics=['accuracy'],
              loss='categorical_crossentropy')

# Train the neural network.
print("\n\n***** TRAINING START ***** ")
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
history = model.fit(datagen.flow(dr.train_X, dr.train_Y, batch_size=64),
                    validation_data=(dr.test_X, dr.test_Y),
                    epochs=EPOCHS,
                    callbacks=[early_stop])
```

- Total params: 4496903 (17.15 MB)
- Trainable params: 4492935 (17.14 MB)
- Non-trainable params: 3968 (15.50 KB)



# EXPERIMENTIAL RESULT



	Precision	Recall	F1 Score
Angry	51%	59%	54%
Disgust	56%	60%	58%
Fear	54%	44%	48%
Happy	<u>84%</u>	<u>84%</u>	<u>84%</u>
Sad	57%	61%	59%
Surprise	53%	47%	50%
Neutral	<u>73%</u>	<u>78%</u>	<u>76%</u>
Accuracy			64%
Macro AVG	61%	62%	61%
Weight AVG	63%	64%	63%

- Accuracy: 64%
- F1 Score : 63%

