**Laporan Machine Learning**

**“Facial Emotion Recognition using CNN”**

James Alberto Ciu / c14190138

Sumano Wijaya Ksatria / c14190159

**Penjelasan Metode dan Dataset**

Pada Proyek mata kuliah *Machine Learning* ini, kami mengambil topik *Facial Emotion Recognition Using CNN,* pada Proyek ini kami menggunakan 2 file yaitu file *train.py* dan *test.py* dengan basis bahasa python. Ada 2 tahapan untuk menjalankan program ini yang pertama adalah menjalankan file *train.py* dan kemudian akan dilakukan *training* menggunakan dataset yang sudah disiapkan, setelah dijalankan file *train.py* kita akan mendapatkan model yang akan digunakan pada saat menjalankan file *test.py*

Saat menjalankan file *train.py* program akan melakukan *training* pada dataset yang berjumlah 57418 foto yang berada didalam folder train. kumpulan foto tersebut dibagi menjadi 7 folder sesuai dengan ekspresi wajah yang terdiri dari *angry, disgusted, fear, happy, neutral, sad, surprise.* setelah dijalankan training program akan melakukan validasi data menggunakan dataset yang berjumlah 14356 foto yang berada didalam folder test.

Pada proyek kali ini kami melakukan 7 percobaan dengan jumlah epoch yang berbeda dan menggunakan 21 foto data lain diluar dataset dengan masing-masing ekspresi 3 foto.

**Dokumentasi Kode Program**

* **Train.py**

**import numpy as np**

**import cv2**

**from keras.models import Sequential**

**from keras.layers import Dense, Dropout, Flatten**

**from keras.layers import Conv2D**

**from tensorflow.keras.optimizers import Adam**

**from keras.layers import MaxPooling2D**

**from keras.preprocessing.image import ImageDataGenerator**

**train\_dir = 'data/train'**

**val\_dir = 'data/test'**

**train\_datagen = ImageDataGenerator(rescale=1./255)**

**val\_datagen = ImageDataGenerator(rescale=1./255)**

**train\_generator = train\_datagen.flow\_from\_directory(**

**train\_dir,**

**target\_size=(48,48),**

**batch\_size=64,**

**color\_mode="grayscale",**

**class\_mode='categorical')**

**validation\_generator = val\_datagen.flow\_from\_directory(**

**val\_dir,**

**target\_size=(48,48),**

**batch\_size=64,**

**color\_mode="grayscale",**

**class\_mode='categorical')**

**emotion\_model = Sequential()**

**emotion\_model.add(Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(48,48,1)))**

**emotion\_model.add(Conv2D(64, kernel\_size=(3, 3), activation='relu'))**

**emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))**

**emotion\_model.add(Dropout(0.25))**

**emotion\_model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu'))**

**emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))**

**emotion\_model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu'))**

**emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))**

**emotion\_model.add(Dropout(0.25))**

**emotion\_model.add(Flatten())**

**emotion\_model.add(Dense(1024, activation='relu'))**

**emotion\_model.add(Dropout(0.5))**

**emotion\_model.add(Dense(7, activation='softmax'))**

**# emotion\_model.load\_weights('emotion\_model.h5')**

**cv2.ocl.setUseOpenCL(False)**

**emotion\_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: "Neutral", 5: "Sad", 6: "Surprised"}**

**emotion\_model.compile(loss='categorical\_crossentropy',optimizer=Adam(lr=0.0001, decay=1e-6),metrics=['accuracy'])**

**emotion\_model\_info = emotion\_model.fit\_generator(**

**train\_generator,**

**steps\_per\_epoch=57418 // 64,**

**epochs=10,**

**validation\_data=validation\_generator,**

**validation\_steps=14356 // 64)**

**emotion\_model.save\_weights('emotion\_model.h5')**

* **Test.py**

**import numpy as np**

**import cv2**

**from keras.models import Sequential**

**from keras.layers import Dense, Dropout, Flatten**

**from keras.layers import Conv2D**

**from tensorflow.keras.optimizers import Adam**

**from keras.layers import MaxPooling2D**

**from keras.preprocessing.image import ImageDataGenerator**

**emotion\_model = Sequential()**

**emotion\_model.add(Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(48,48,1)))**

**emotion\_model.add(Conv2D(64, kernel\_size=(3, 3), activation='relu'))**

**emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))**

**emotion\_model.add(Dropout(0.25))**

**emotion\_model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu'))**

**emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))**

**emotion\_model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu'))**

**emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))**

**emotion\_model.add(Dropout(0.25))**

**emotion\_model.add(Flatten())**

**emotion\_model.add(Dense(1024, activation='relu'))**

**emotion\_model.add(Dropout(0.5))**

**emotion\_model.add(Dense(7, activation='softmax'))**

**emotion\_model.load\_weights('emotion\_model.h5')**

**cv2.ocl.setUseOpenCL(False)**

**emotion\_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: "Neutral", 5: "Sad", 6: "Surprised"}**

**# start the webcam feed**

**cap = cv2.VideoCapture(0)**

**while True:**

**# Find haar cascade to draw bounding box around face**

**ret, frame = cap.read()**

**if not ret:**

**break**

**bounding\_box = cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml")**

**gray\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)**

**num\_faces = bounding\_box.detectMultiScale(gray\_frame,scaleFactor=1.3, minNeighbors=5)**

**for (x, y, w, h) in num\_faces:**

**cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (255, 0, 0), 2)**

**roi\_gray\_frame = gray\_frame[y:y + h, x:x + w]**

**cropped\_img = np.expand\_dims(np.expand\_dims(cv2.resize(roi\_gray\_frame, (48, 48)), -1), 0)**

**emotion\_prediction = emotion\_model.predict(cropped\_img)**

**maxindex = int(np.argmax(emotion\_prediction))**

**cv2.putText(frame, emotion\_dict[maxindex], (x+20, y-60), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE\_AA)**

**cv2.imshow('Video', cv2.resize(frame,(1200,860),interpolation = cv2.INTER\_CUBIC))**

**if cv2.waitKey(1) & 0xFF == ord('q'):**

**break**

**cap.release()**

**cv2.destroyAllWindows()**

**Hasil dan Analisa**

* Inputan:

21 foto dari luar dataset:

* angry



* disgust



* fear



* happy



* neutral



* sad

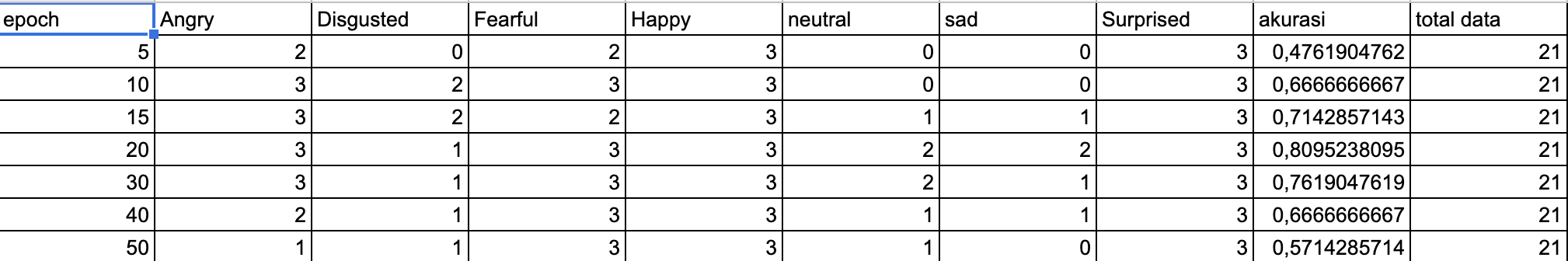


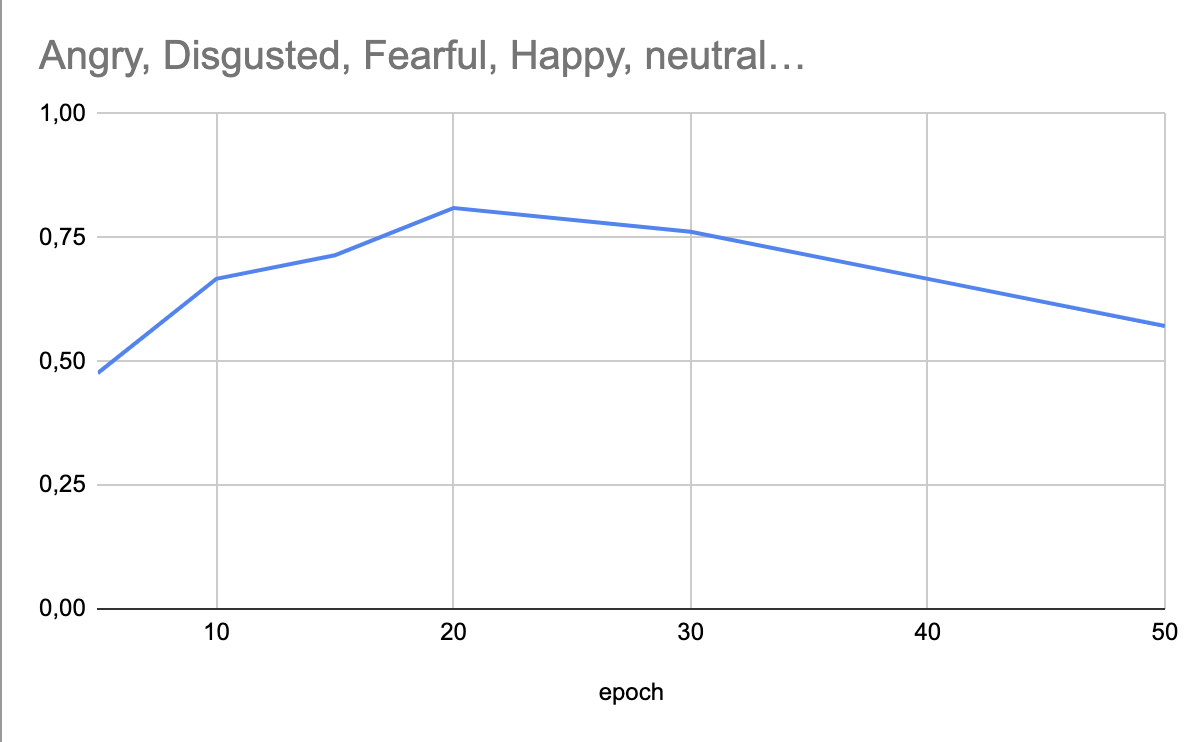


* surprise



* percobaan dengan mengubah jumlah epoch





**Kesimpulan**

Kesimpulan yang kami dapat adalah bahwa pada awal percobaan kami mendapatkan hasil akurasi yang meningkat pada percobaan epoch 5-20 dengan interval 5 namun setelah dari epoch 20-50 dengan interval 10 kami mendapatkan bahwa akurasi menurun kembali hal ini dikarenakan error yang dihasilkan juga ikut meningkat. Sebelum mencoba mengubah epoch kami juga sempat mencoba beberapa dataset dan pada awalnya kami mendapatkan akurasi yang sangat rendah dari dataset tersebut namun setelah kami menambah dataset kami mendapatkan hasil yang lebih bagus.