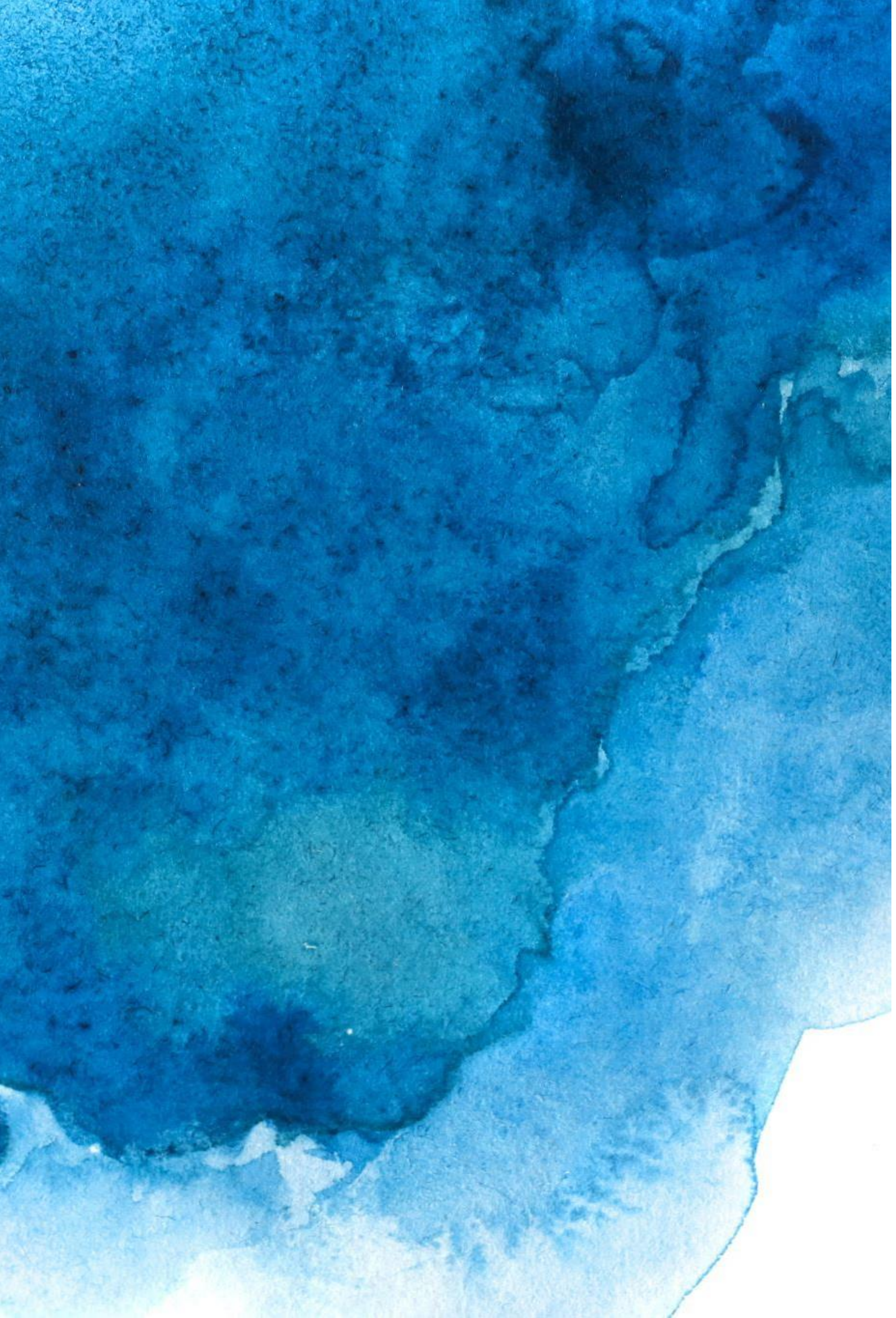
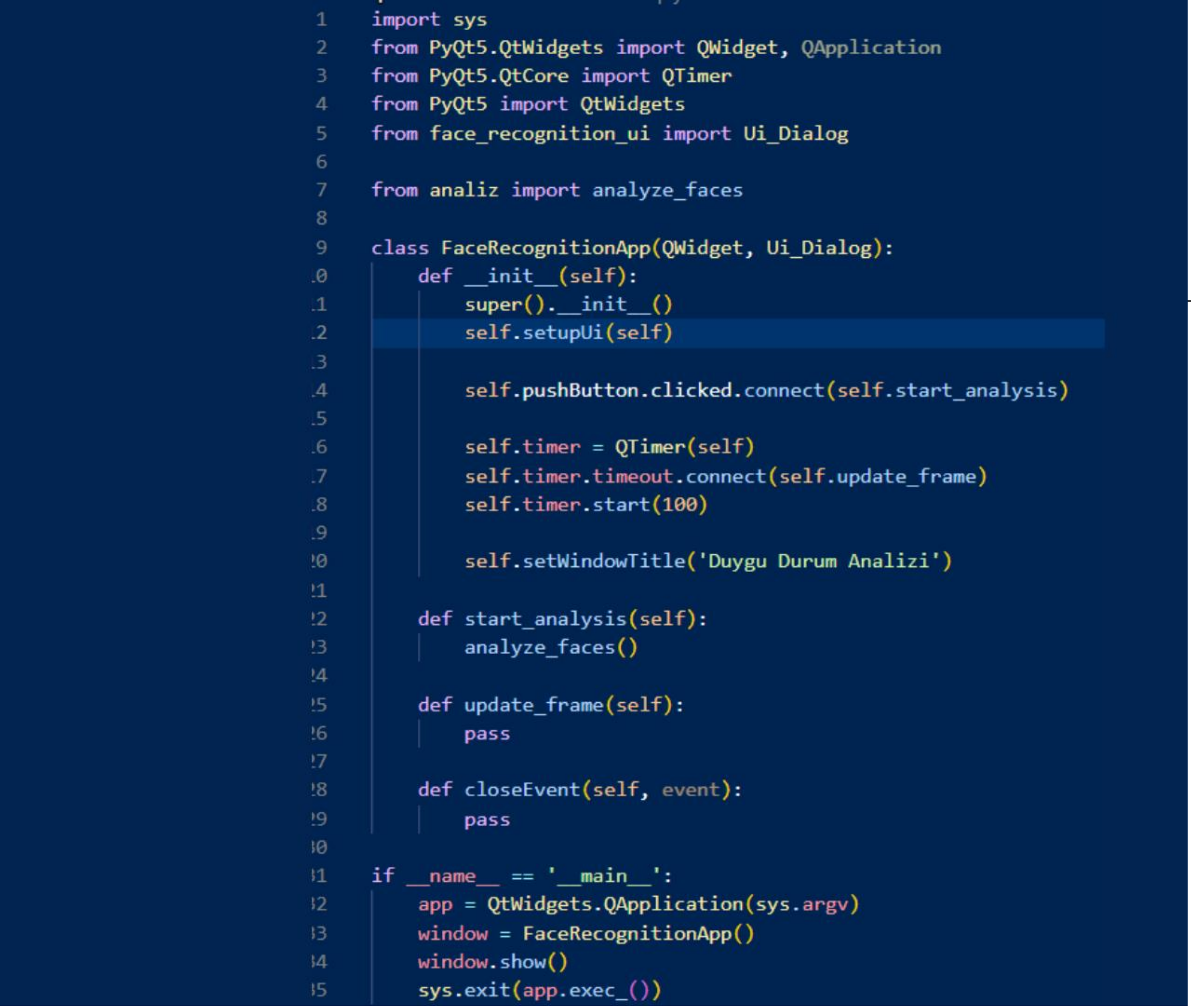


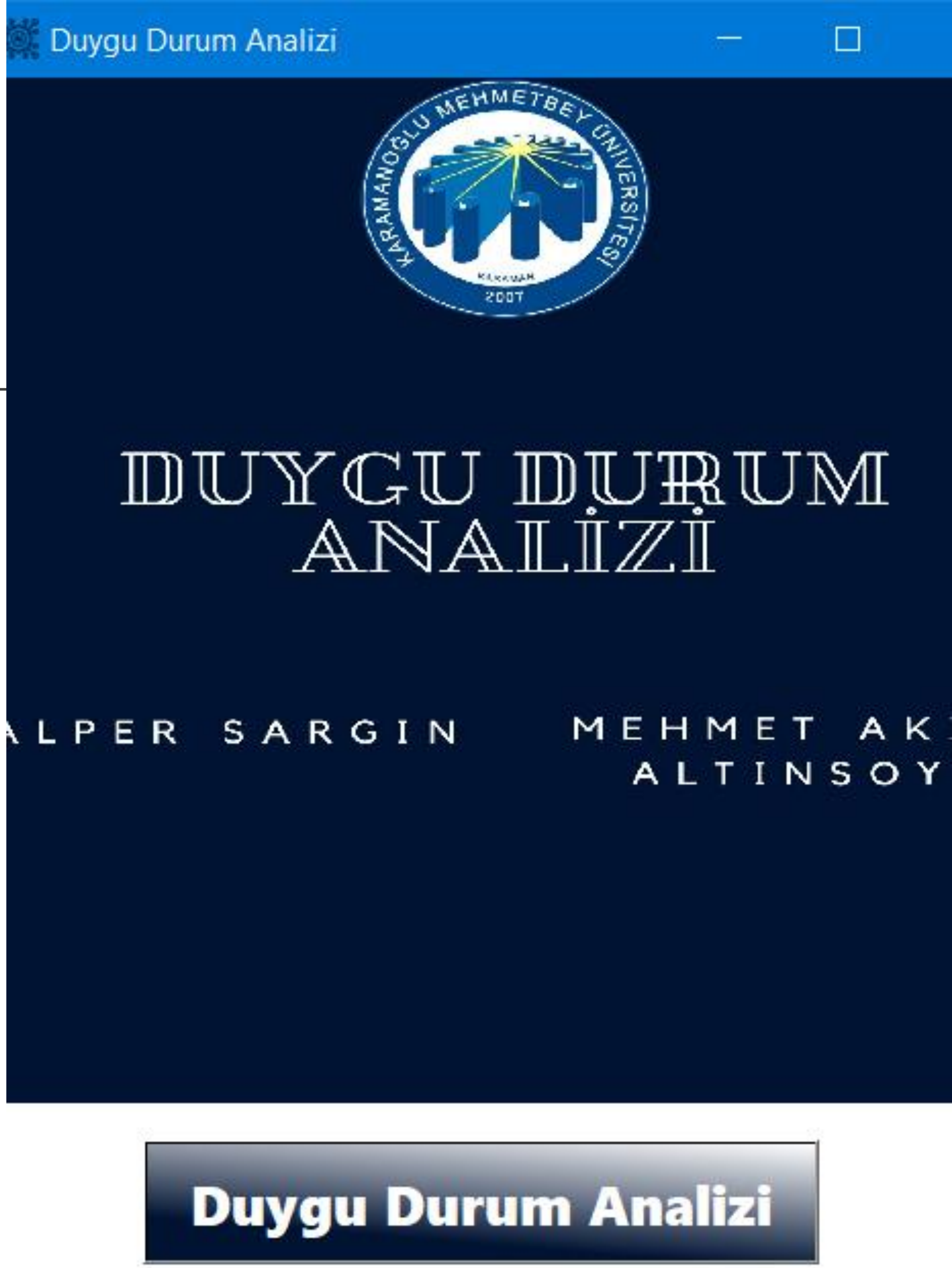
Duygu Durum Analizi

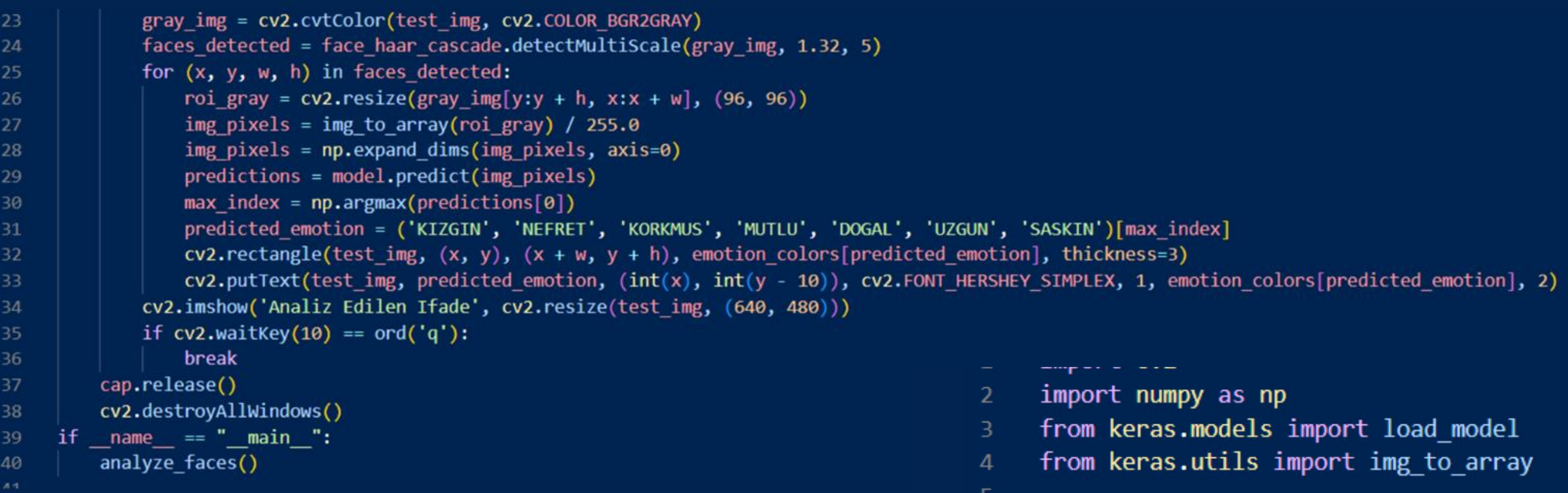
MEHMET AKIF ALTıNSOY

ALPER SARGıN

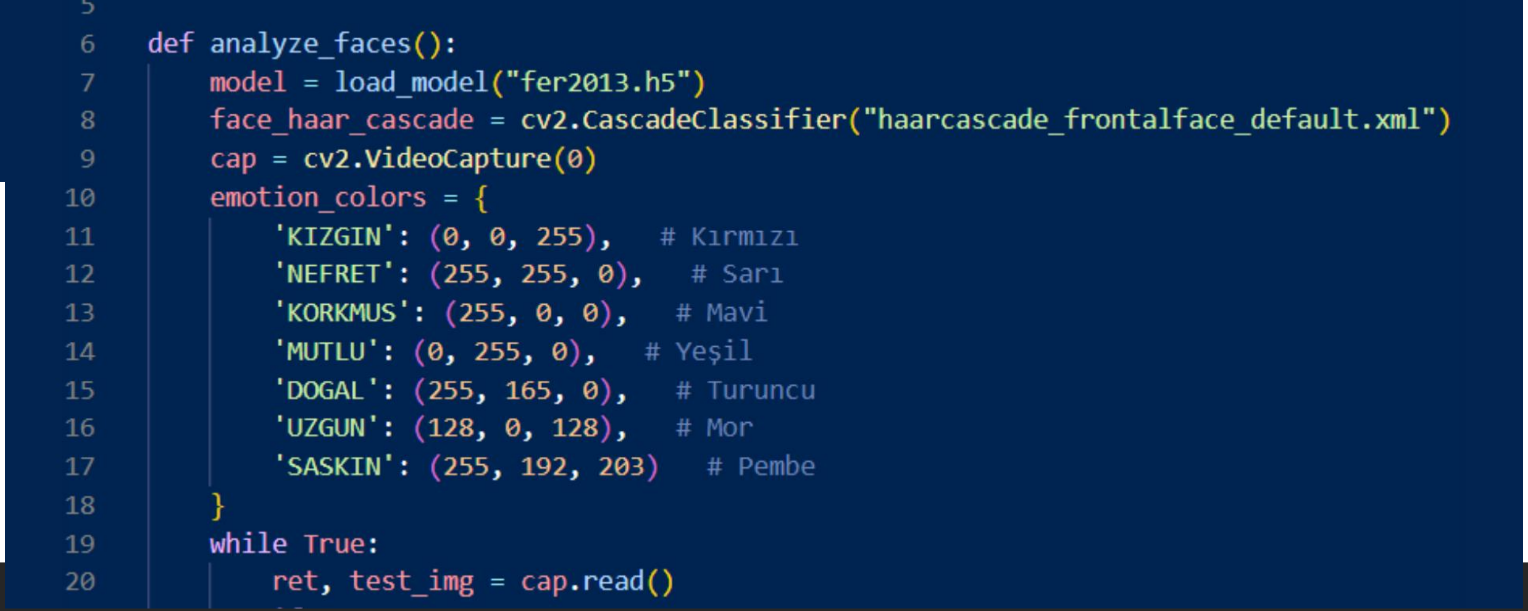


```python
import sys
from PyQt5.QtWidgets import QWidget, QApplication
from PyQt5.QtCore import QTimer
from PyQt5 import QtWidgets
from face_recognition_ui import Ui_Dialog

from analiz import analyze_faces

class FaceRecognitionApp(QWidget, Ui_Dialog):
    def __init__(self):
        super().__init__()
        self.setupUi(self)

        self.pushButton.clicked.connect(self.start_analysis)

        self.timer = QTimer(self)
        self.timer.timeout.connect(self.update_frame)
        self.timer.start(100)

        self.setWindowTitle('Duygu Durum Analizi')

    def start_analysis(self):
        analyze_faces()

    def update_frame(self):
        pass

    def closeEvent(self, event):
        pass

if __name__ == '__main__':
    app = QtWidgets.QApplication(sys.argv)
    window = FaceRecognitionApp()
    window.show()
    sys.exit(app.exec_())
```



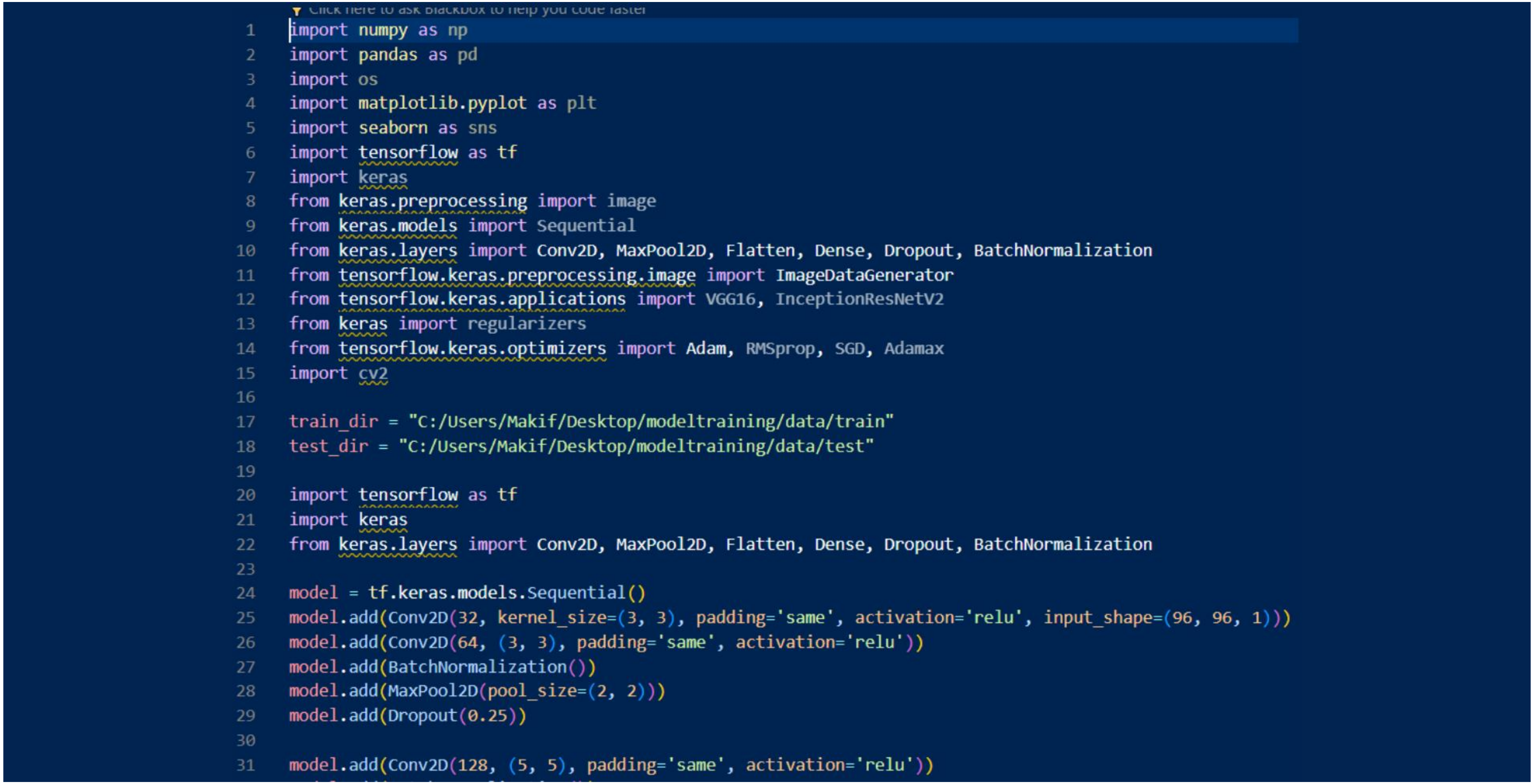


```python
        gray_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2GRAY)
        faces_detected = face_haar_cascade.detectMultiScale(gray_img, 1.32, 5)
        for (x, y, w, h) in faces_detected:
            roi_gray = cv2.resize(gray_img[y:y + h, x:x + w], (96, 96))
            img_pixels = img_to_array(roi_gray) / 255.0
            img_pixels = np.expand_dims(img_pixels, axis=0)
            predictions = model.predict(img_pixels)
            max_index = np.argmax(predictions[0])
            predicted_emotion = ('KIZGIN', 'NEFRET', 'KORKMUS', 'MUTLU', 'DOGAL', 'UZGUN', 'SASKIN')[max_index]
            cv2.rectangle(test_img, (x, y), (x + w, y + h), emotion_colors[predicted_emotion], thickness=3)
            cv2.putText(test_img, predicted_emotion, (int(x), int(y - 10)), cv2.FONT_HERSHEY_SIMPLEX, 1, emotion_colors[predicted_emotion], 2)
        cv2.imshow('Analiz Edilen Ifade', cv2.resize(test_img, (640, 480)))
        if cv2.waitKey(10) == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
if __name__ == "__main__":
    analyze_faces()
```



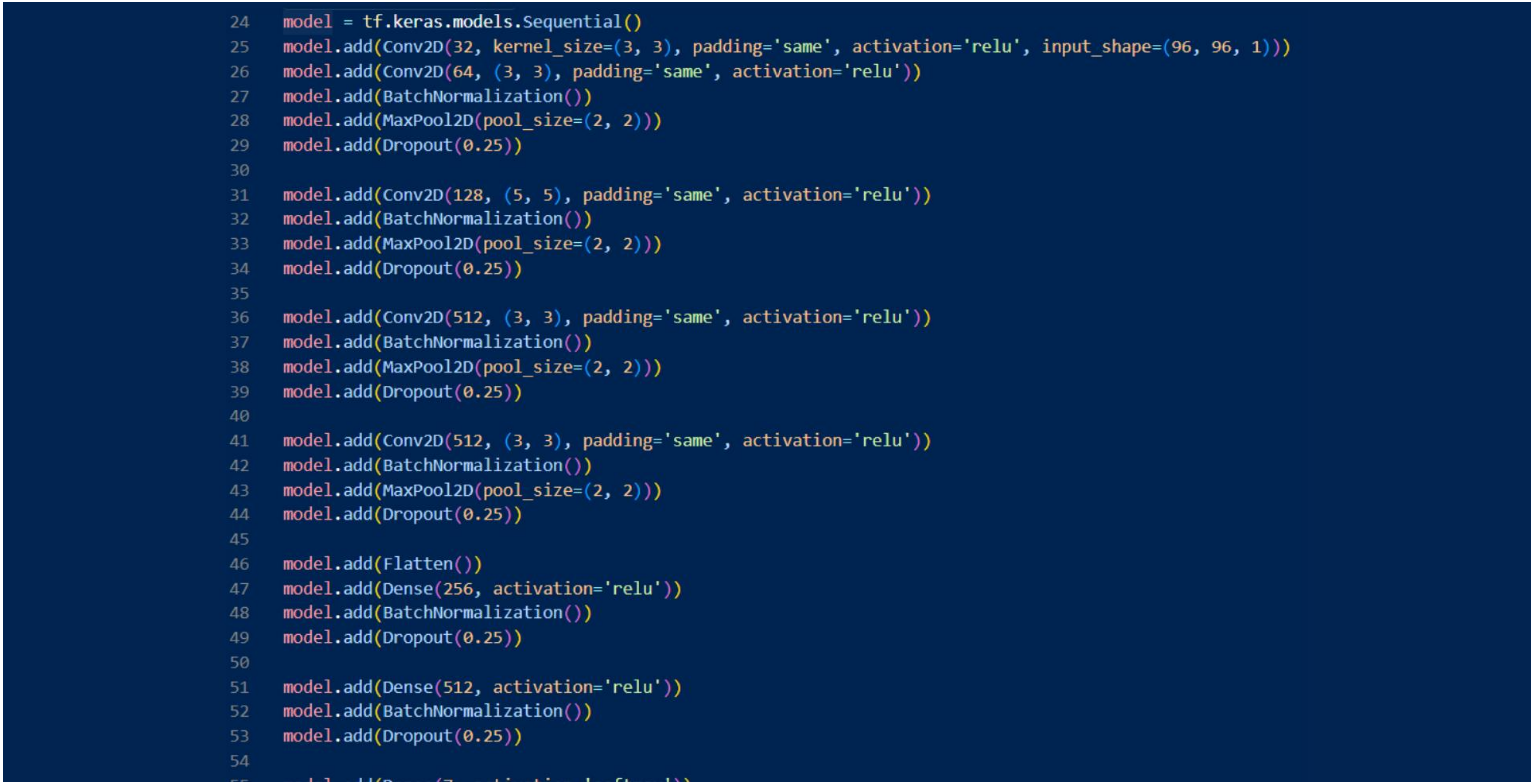
```python
import numpy as np
from keras.models import load_model
from keras.utils import img_to_array

def analyze_faces():
    model = load_model("fer2013.h5")
    face_haar_cascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
    cap = cv2.VideoCapture(0)
    emotion_colors = {
        'KIZGIN': (0, 0, 255),    # Kırmızı
        'NEFRET': (255, 255, 0),   # Sarı
        'KORKMUS': (255, 0, 0),   # Mavi
        'MUTLU': (0, 255, 0),   # Yeşil
        'DOGAL': (255, 165, 0),   # Turuncu
        'UZGUN': (128, 0, 128),   # Mor
        'SASKIN': (255, 192, 203)   # Pembe
    }
    while True:
        ret, test_img = cap.read()
```

Model Eğitimi

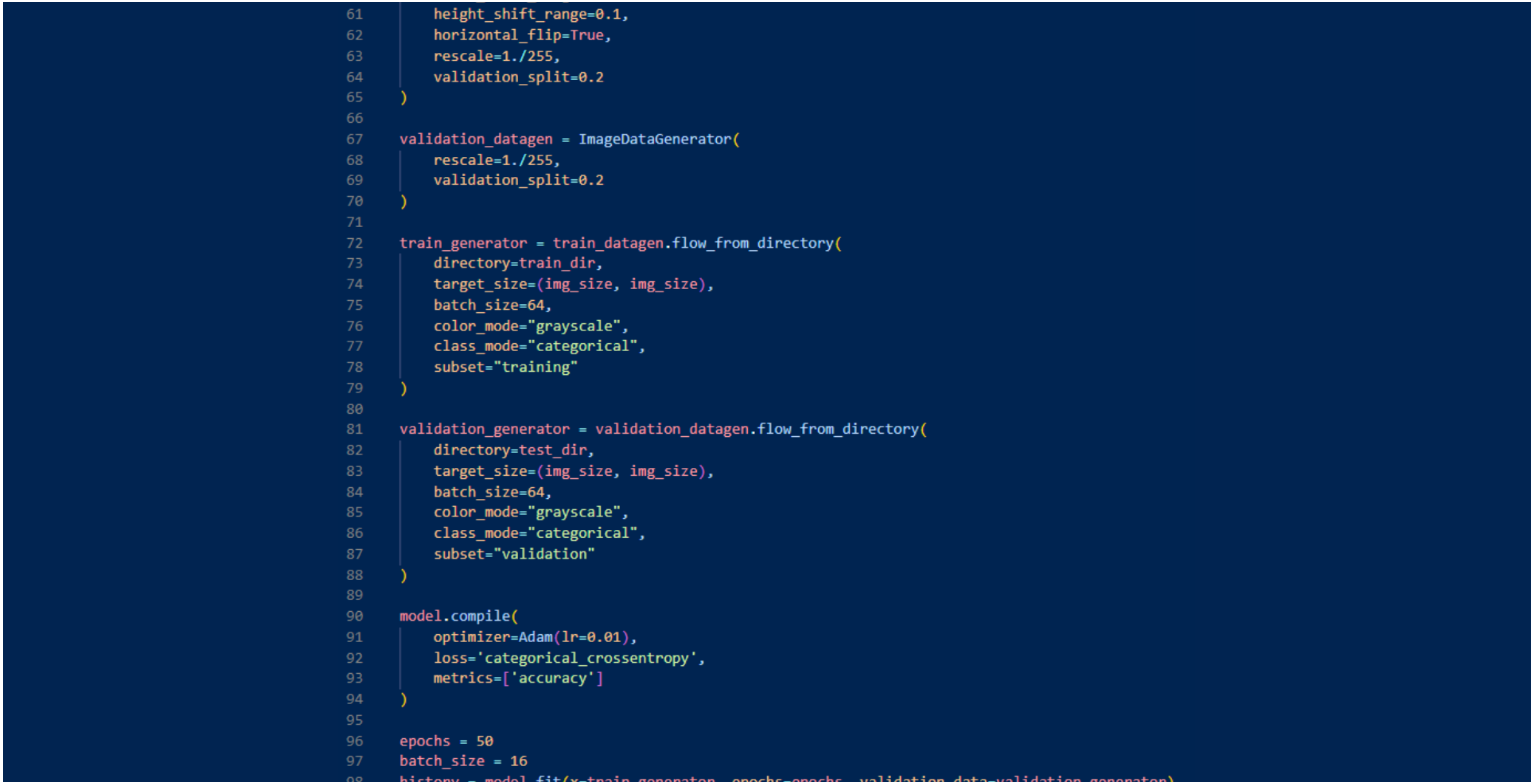


Click here to ask Blackbox to help you code faster

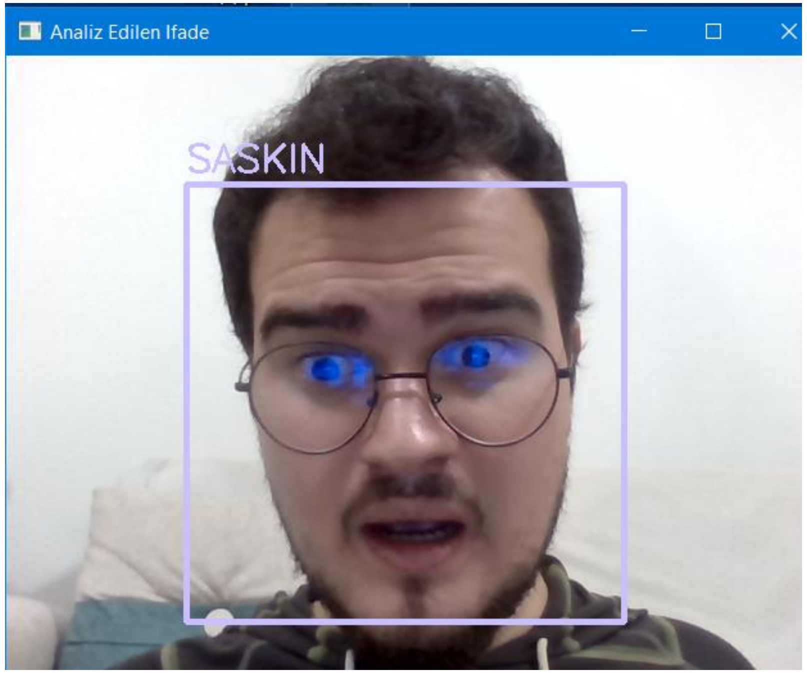
```python
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
import keras
from keras.preprocessing import image
from keras.models import Sequential
from keras.layers import Conv2D, MaxPool2D, Flatten, Dense, Dropout, BatchNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16, InceptionResNetV2
from keras import regularizers
from tensorflow.keras.optimizers import Adam, RMSprop, SGD, Adamax
import cv2

train_dir = "C:/Users/Makif/Desktop/modeltraining/data/train"
test_dir = "C:/Users/Makif/Desktop/modeltraining/data/test"

import tensorflow as tf
import keras
from keras.layers import Conv2D, MaxPool2D, Flatten, Dense, Dropout, BatchNormalization

model = tf.keras.models.Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), padding='same', activation='relu', input_shape=(96, 96, 1)))
model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (5, 5), padding='same', activation='relu'))
```

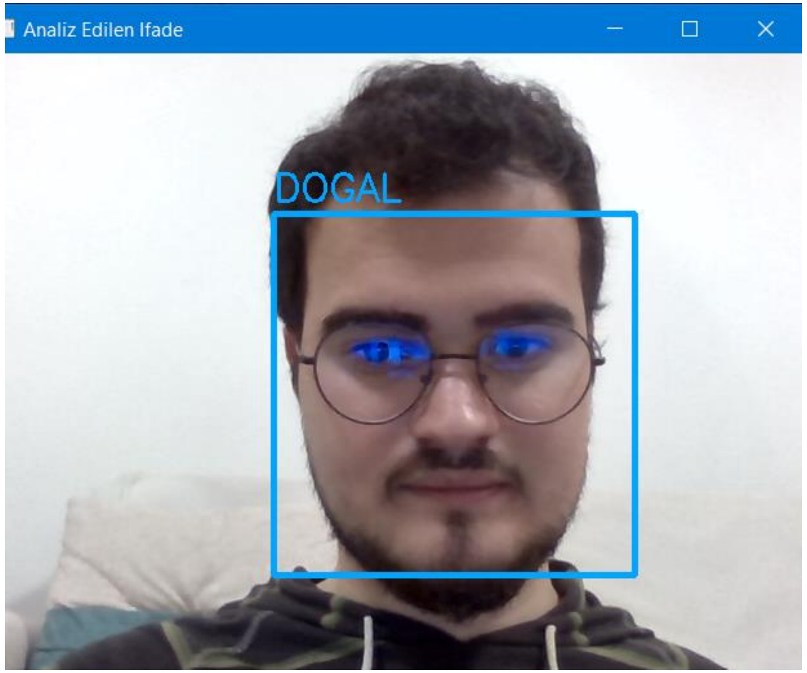


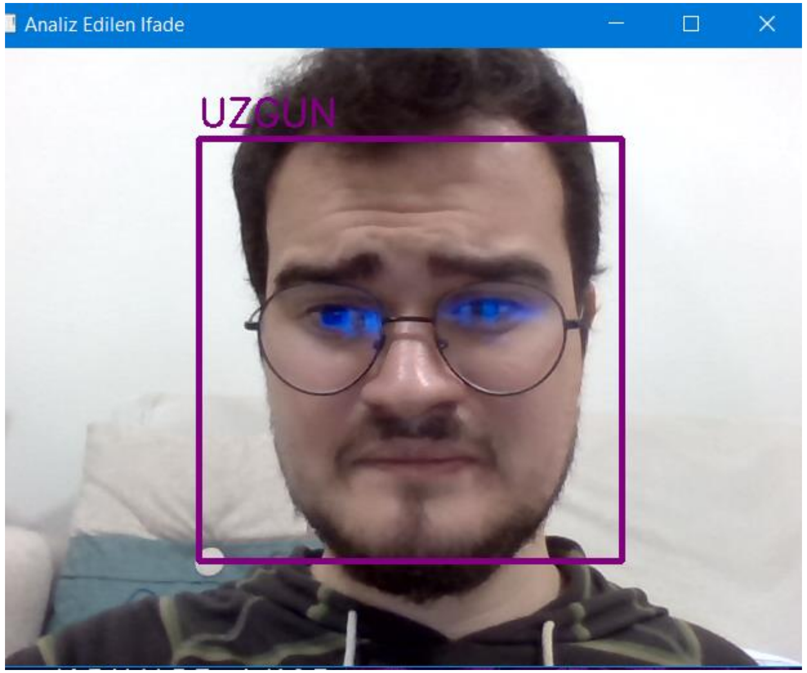
```python
model = tf.keras.models.Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), padding='same', activation='relu', input_shape=(96, 96, 1)))
model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (5, 5), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(512, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(512, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.25))

model.add(Dense(512, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.25))

model.add(Dense(7, activation='softmax'))
```

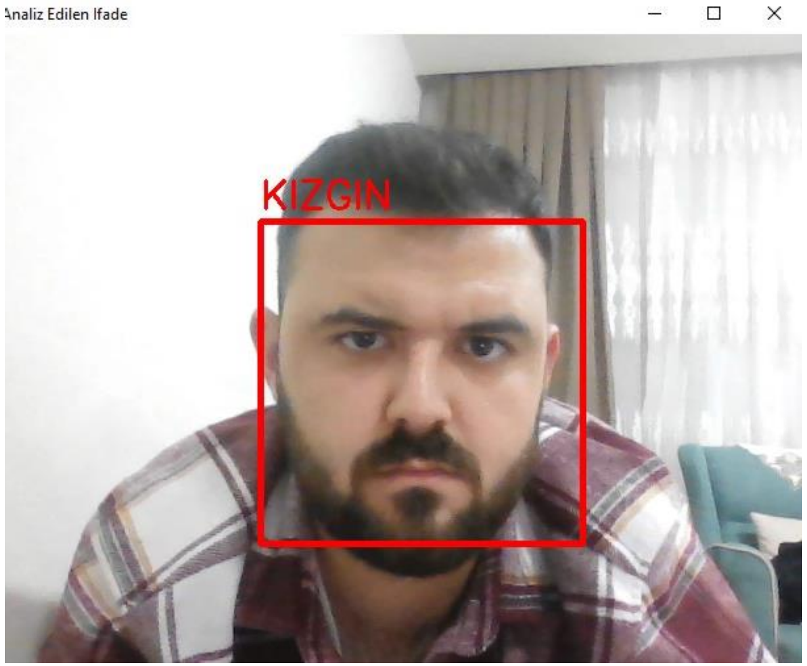


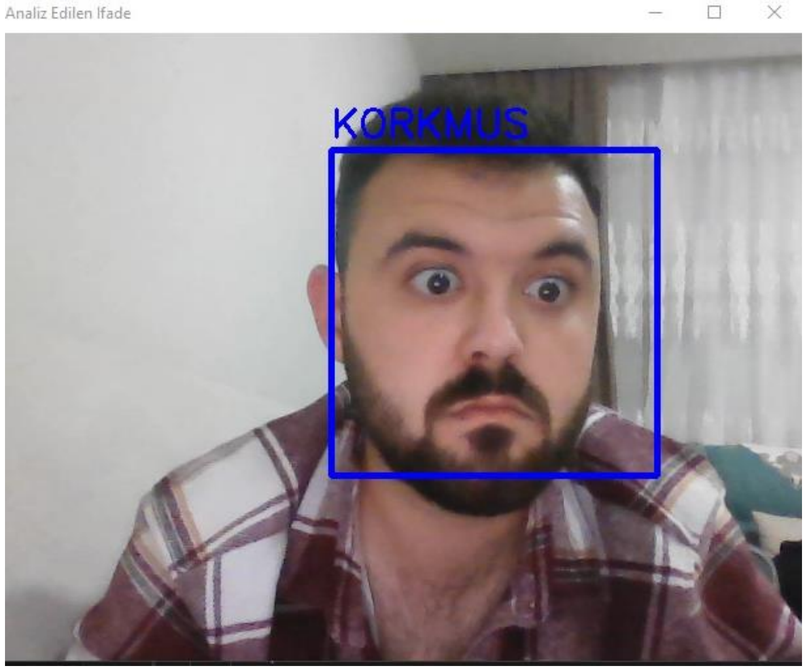
```python
    height_shift_range=0.1,
    horizontal_flip=True,
    rescale=1./255,
    validation_split=0.2
)

validation_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2
)

train_generator = train_datagen.flow_from_directory(
    directory=train_dir,
    target_size=(img_size, img_size),
    batch_size=64,
    color_mode="grayscale",
    class_mode="categorical",
    subset="training"
)

validation_generator = validation_datagen.flow_from_directory(
    directory=test_dir,
    target_size=(img_size, img_size),
    batch_size=64,
    color_mode="grayscale",
    class_mode="categorical",
    subset="validation"
)

model.compile(
    optimizer=Adam(lr=0.01),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

epochs = 50
batch_size = 16
history = model.fit(x=train_generator, epochs=epochs, validation_data=validation_generator)
```

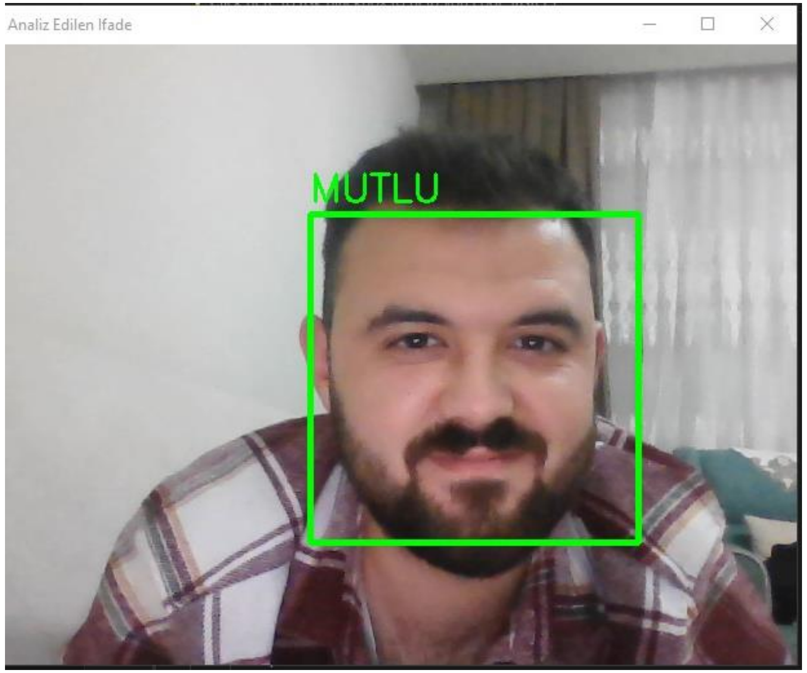

Analiz Edilen Ifade
SASKIN

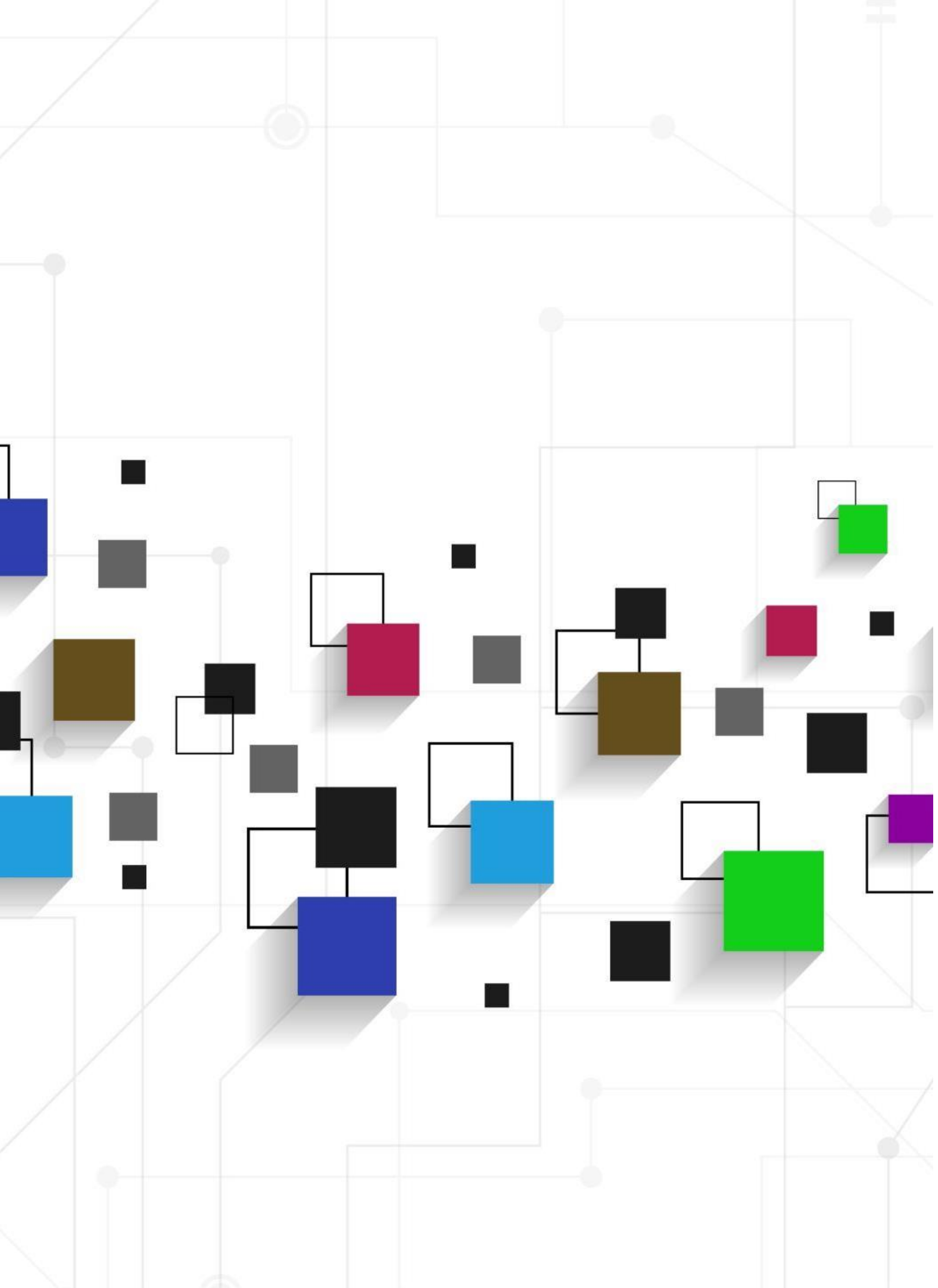

Analiz Edilen Ifade
DOGAL


Analiz Edilen Ifade
UZGUN


Analiz Edilen Ifade
KIZGIN


Analiz Edilen Ifade
KORKMUS


Analiz Edilen Ifade
MUTLU



Bizi Dinlediğiniz İçin Teşekkürler