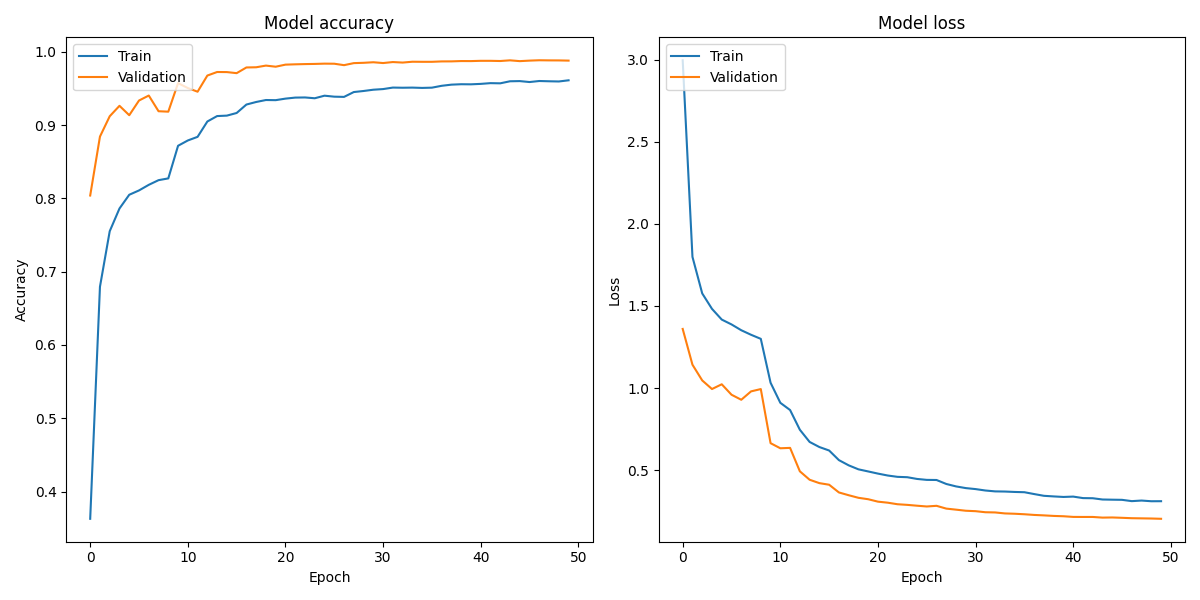
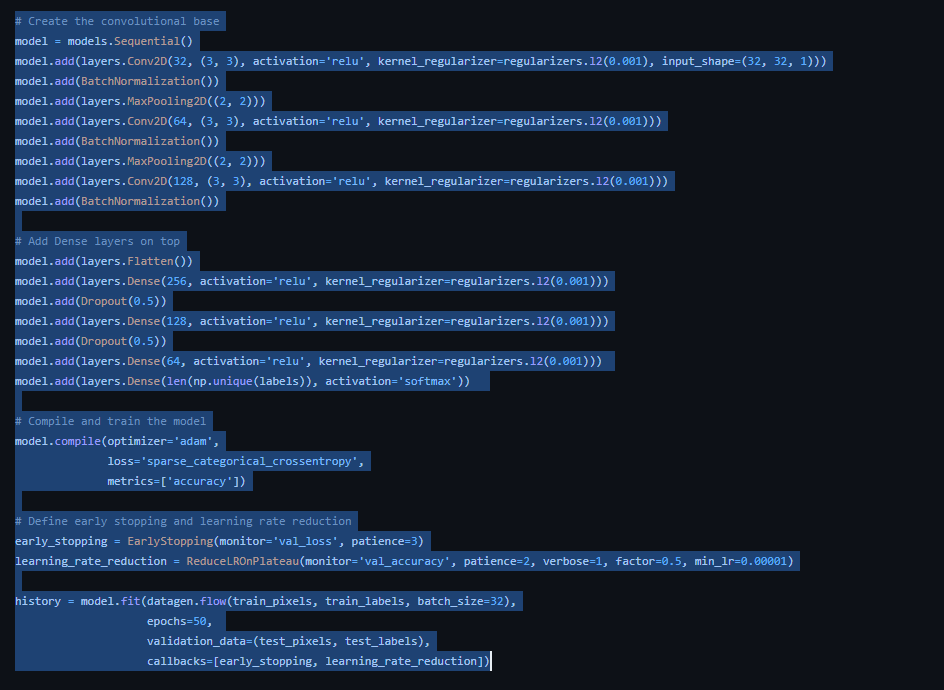
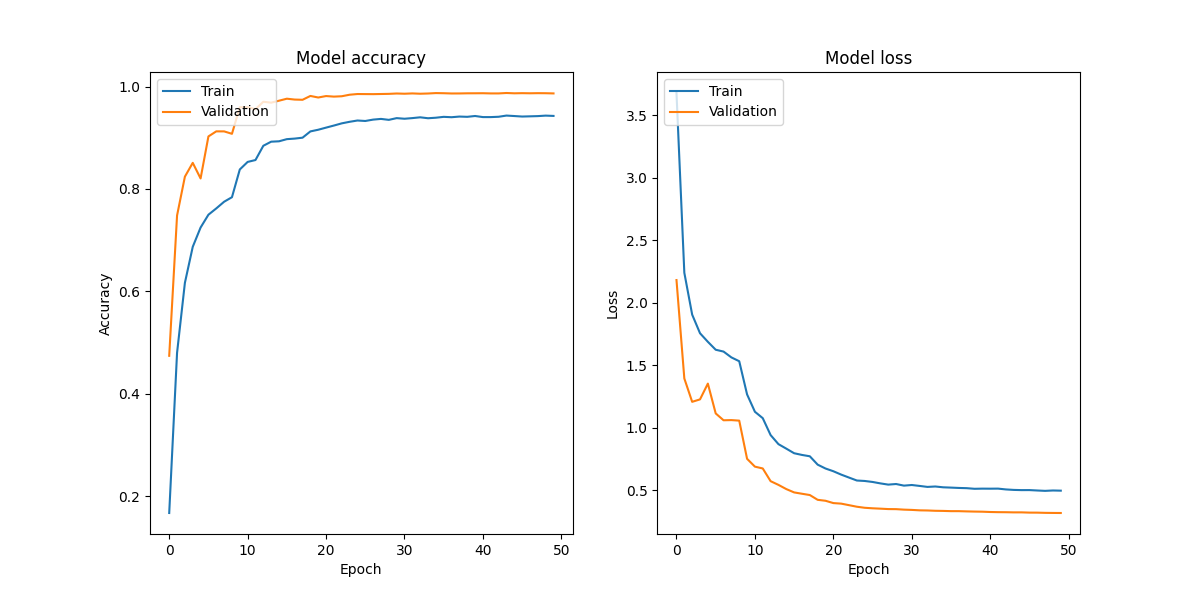
Training 1:



Condition of over fit occurred here.



Training 2:



# Create the convolutional base

model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu', kernel\_regularizer=regularizers.l2(0.001), input\_shape=(32, 32, 1)))

model.add(BatchNormalization())

model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu', kernel\_regularizer=regularizers.l2(0.001)))

model.add(BatchNormalization())

model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(128, (3, 3), activation='relu', kernel\_regularizer=regularizers.l2(0.001)))

model.add(BatchNormalization())

model.add(layers.Conv2D(128, (3, 3), activation='relu', kernel\_regularizer=regularizers.l2(0.001)))

model.add(BatchNormalization())

# Add Dense layers on top

model.add(layers.Flatten())

model.add(layers.Dense(256, activation='relu', kernel\_regularizer=regularizers.l2(0.001)))

model.add(Dropout(0.5))

model.add(layers.Dense(128, activation='relu', kernel\_regularizer=regularizers.l2(0.001)))

model.add(Dropout(0.5))

model.add(layers.Dense(64, activation='relu', kernel\_regularizer=regularizers.l2(0.001)))

model.add(Dropout(0.5))

model.add(layers.Dense(len(np.unique(labels)), activation='softmax'))

# Compile and train the model

model.compile(optimizer='adam',

              loss='sparse\_categorical\_crossentropy',

              metrics=['accuracy'])

# Define early stopping and learning rate reduction

early\_stopping = EarlyStopping(monitor='val\_loss', patience=3)

learning\_rate\_reduction = ReduceLROnPlateau(monitor='val\_accuracy', patience=2, verbose=1, factor=0.5, min\_lr=0.00001)

history = model.fit(datagen.flow(train\_pixels, train\_labels, batch\_size=32),

                    epochs=50,

                    validation\_data=(test\_pixels, test\_labels),

                    callbacks=[early\_stopping, learning\_rate\_reduction])