

Cílem mého projektu bylo použít natrénovaný OCR model na webové stránce.

Začal jsem tím, že jsem natrénoval model na datech z IAM Handwriting Database s pomocí tutoriálu (1). Model měl 8 skrytých vrstev s aktivační funkcí leaky Relu a na konci jednu vrstvu s aktivační funkcí softmax.

```
def train_model(input_dim, output_dim, activation="leaky_relu", dropout=0.2):  
    inputs = layers.Input(shape=input_dim, name="input")  
    # normalize images here instead in preprocessing step  
    input = layers.Lambda(lambda x: x / 255)(inputs)  
    x1 = residual_block(input, 16, activation=activation, skip_conv=True, strides=1, dropout=dropout)  
    x2 = residual_block(x1, 16, activation=activation, skip_conv=True, strides=2, dropout=dropout)  
    x3 = residual_block(x2, 16, activation=activation, skip_conv=False, strides=1, dropout=dropout)  
    x4 = residual_block(x3, 32, activation=activation, skip_conv=True, strides=2, dropout=dropout)  
    x5 = residual_block(x4, 32, activation=activation, skip_conv=False, strides=1, dropout=dropout)  
    x6 = residual_block(x5, 64, activation=activation, skip_conv=True, strides=2, dropout=dropout)  
    x7 = residual_block(x6, 64, activation=activation, skip_conv=True, strides=1, dropout=dropout)  
    x8 = residual_block(x7, 64, activation=activation, skip_conv=False, strides=1, dropout=dropout)  
    x9 = residual_block(x8, 64, activation=activation, skip_conv=False, strides=1, dropout=dropout)  
    squeezed = layers.Reshape((x9.shape[-3] * x9.shape[-2], x9.shape[-1]))(x9)  
    blstm = layers.Bidirectional(layers.LSTM(128, return_sequences=True))(squeezed)  
    blstm = layers.Dropout(dropout)(blstm)  
    output = layers.Dense(output_dim + 1, activation="softmax", name="output")(blstm)  
    model = Model(inputs=inputs, outputs=output)  
    return model
```

Před trénováním se všechny obrázky musely resizenout na stejnou velikost. Model jsem uložil ve formátu HDF5. Dále jsem chtěl použít tensorflow.js abych deploynul svůj model na webovou stránku, tam nastala moje největší potíž. Aby se dal model deploynout pomocí tensorflow.js, musí být v .tfjs formátu, můj model je ve formátu .h5. Tento problém jsem se pokusil vyřešit tfjs funkcí converter, která slibovala přesně to, co jsem potřeboval. Když jsem se však pokusil touto cestou svůj model převést, program skončil chybou.

Nepovedlo se mi upravit program tak, aby chybou nekončil, tak jsem zkusil použít jiný model, tentokrát jednoduchý digit recognition model s jednou flatten vrstvou, dvěma relu vrstvami, a nakonec softmax vrstvou.

```
model = tf.keras.models.Sequential()  
model.add(tf.keras.layers.Flatten())  
model.add(tf.keras.layers.Dense(units=128, activation=tf.nn.relu))  
model.add(tf.keras.layers.Dense(units=128, activation=tf.nn.relu))  
model.add(tf.keras.layers.Dense(units=10, activation=tf.nn.softmax))
```

Když jsem se pokusil převést tento model, dostal jsem tu samou chybu, tak jsem začal pátrat po tom, co ji způsobuje. Zjistil jsem, že chyba je někde v tf decision forests. Zkusil jsem tensorflow přeinstalovat, ale stále jsem dostával tu samou chybu. Nakonec jsem na Stackoverflow zjistil, že tensorflow decision forest package nepodporuje Windows.

Tak jsem hledal alternativu, která nebude používat tensorflow.js a nejjednodušší alternativa, kterou se mi podařilo zprovoznit je použít tesseract.js.

(1) https://youtu.be/WhRC31SlXzA?si=Ysy0uW_RlyAbFh_I