Sign Language Fingerspelling Recognition with TensorFlow

"This notebook guides you through the process of training a Transformer model using TensorFlow on the Google American Sign Language Fingerspelling Recognition dataset.

The objective of the model is to predict and translate American Sign Language (ASL) fingerspelling from a set of video frames into text (a 'phrase').

In this notebook, you will learn:

How to load the data Convert the data to TFRecords to expedite model retraining Train Transformer models on the data Convert the model to TFLite"

Import the libraries

In [2]:
```
import os
import shutil
import numpy as np
import pandas as pd
import pyarrow.parquet as pq
import tensorflow as tf
import json
import mediapipe
import matplotlib
import matplotlib.pyplot as plt
import random

from skimage.transform import resize
from mediapipe.framework.formats import landmark_pb2
from tensorflow import keras
from tensorflow.keras import layers
from tqdm.notebook import tqdm
from matplotlib import animation, rc
```

```
/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/__init__.py:98: User
Warning: unable to load libtensorflow_io_plugins.so: unable to open file: libtensorfl
ow_io_plugins.so, from paths: ['/opt/conda/lib/python3.10/site-packages/tensorflow_i
o/python/ops/libtensorflow_io_plugins.so']
caused by: ['/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/libtens
orflow_io_plugins.so: undefined symbol: _ZN3tsl6StatusC1EN10tensorflow5error4CodeESt1
7basic_string_viewIcSt11char_traitsIcEENS_14SourceLocationE']
  warnings.warn(f"unable to load libtensorflow_io_plugins.so: {e}")
/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/__init__.py:104: Use
rWarning: file system plugins are not loaded: unable to open file: libtensorflow_io.s
o, from paths: ['/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/lib
tensorflow_io.so']
caused by: ['/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/libtens
orflow_io.so: undefined symbol: _ZTVN10tensorflow13GcsFileSystemE']
  warnings.warn(f"file system plugins are not loaded: {e}")
```

In [3]:
```
print("TensorFlow v" + tf.__version__)
print("Mediapipe v" + mediapipe.__version__)
```

```
TensorFlow v2.12.0
Mediapipe v0.10.7
```

Load the Dataset

In [4]:
```
dataset_df = pd.read_csv('/input/asl-fingerspelling/train.csv')
print("Full train dataset shape is {}".format(dataset_df.shape))
```

```
Full train dataset shape is (67208, 5)
```

In [5]:
```
dataset_df.head()
```

Out[5]:

	path	file_id	sequence_id	participant_id	phrase
0	train_landmarks/5414471.parquet	5414471	1816796431	217	3 creekhouse
1	train_landmarks/5414471.parquet	5414471	1816825349	107	scales/kuhaylah
2	train_landmarks/5414471.parquet	5414471	1816909464	1	1383 william lanier
3	train_landmarks/5414471.parquet	5414471	1816967051	63	988 franklin lane
4	train_landmarks/5414471.parquet	5414471	1817123330	89	6920 northeast 661st road

Quick basic dataset exploration

Each entry in train.csv contains a phrase, its sequence_id, path, and file_id. The file_id indicates the file that holds the landmarks data for that particular phrase, and sequence_id is the unique index of the landmark sequence within the landmarks data file.

Examine the landmarks file for the first row of `train.csv`.

In [6]:
```
# Fetch sequence_id, file_id, phrase from first row
sequence_id, file_id, phrase = dataset_df.iloc[0][['sequence_id', 'file_id', 'phrase']
print(f"sequence_id: {sequence_id}, file_id: {file_id}, phrase: {phrase}")
```

```
sequence_id: 1816796431, file_id: 5414471, phrase: 3 creekhouse
```

Open the parquet file and fetch the data for the particular `sequence_id`.

In [7]:
```
# Fetch data from parquet file
sample_sequence_df = pq.read_table(f"/input/asl-fingerspelling/train_landmarks/{str(fi
    filters=[[('sequence_id', '=', sequence_id)],]).to_pandas()
print("Full sequence dataset shape is {}".format(sample_sequence_df.shape))
```

```
Full sequence dataset shape is (123, 1630)
```

This particular phrase(***3 creekhouse***) contains 123 entries (or frames) and 1630 columns including 1628 landmark coordinates.

In [8]:
```
sample_sequence_df.head()
```

Out[8]:

sequence_id	frame	x_face_0	x_face_1	x_face_2	x_face_3	x_face_4	x_face_5	x_face_6	x_face_7	x
1816796431	0	0.710588	0.699951	0.705657	0.691768	0.699669	0.701980	0.709724	0.610405	0
1816796431	1	0.709525	0.697582	0.703713	0.691016	0.697576	0.700467	0.709796	0.616540	0
1816796431	2	0.711059	0.700858	0.706272	0.693285	0.700825	0.703319	0.711549	0.615606	0
1816796431	3	0.712799	0.702518	0.707840	0.694899	0.702445	0.704794	0.712483	0.625044	0
1816796431	4	0.712349	0.705451	0.709918	0.696006	0.705180	0.706928	0.712685	0.614356	0

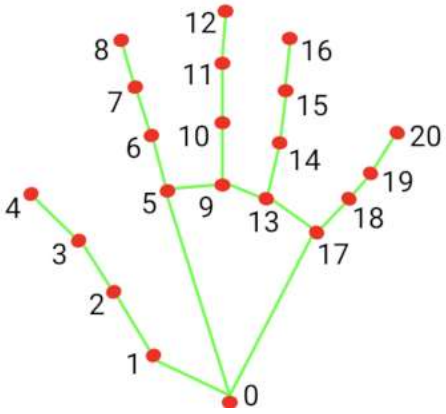
5 rows × 1630 columns

Data visualization using mediapipe APIs

Hand landmarks represent the key points on a human hand.

In [9]:
```
# Function create animation from images.

matplotlib.rcParams['animation.embed_limit'] = 2**128
matplotlib.rcParams['savefig.pad_inches'] = 0
rc('animation', html='jshtml')

def create_animation(images):
    fig = plt.figure(figsize=(6, 9))
    ax = plt.Axes(fig, [0., 0., 1., 1.])
    ax.set_axis_off()
    fig.add_axes(ax)
    im=ax.imshow(images[0], cmap="gray")
    plt.close(fig)

    def animate_func(i):
        im.set_array(images[i])
        return [im]

    return animation.FuncAnimation(fig, animate_func, frames=len(images), interval=100
```

This image illustrates the 21 keypoints on the hand.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

source: https://developers.google.com/mediapipe/solutions/vision/hand_landmarker

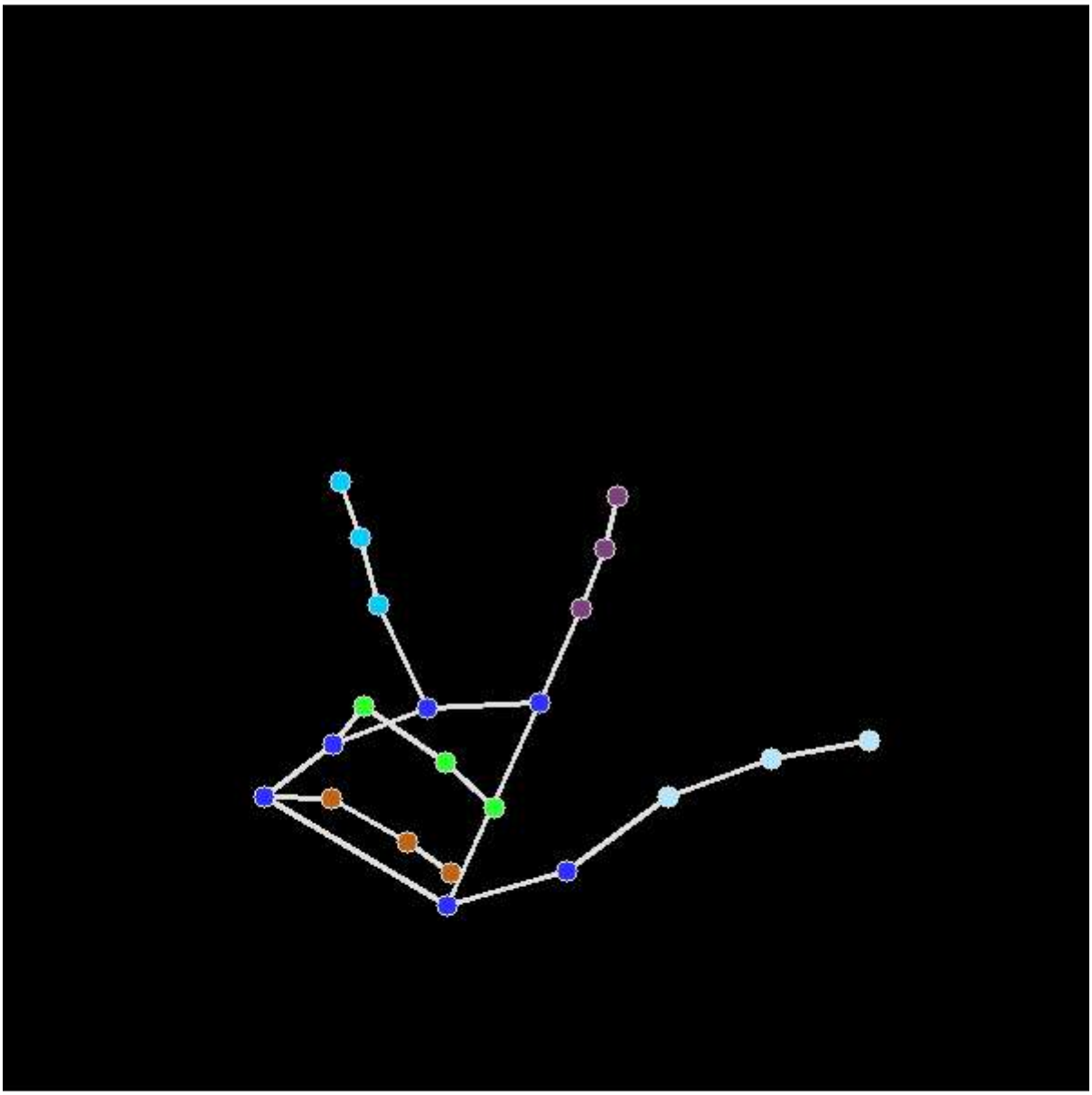
In [10]:
```
# This function extracts the data for both hands.

mp_pose = mediapipe.solutions.pose
mp_hands = mediapipe.solutions.hands
mp_drawing = mediapipe.solutions.drawing_utils
mp_drawing_styles = mediapipe.solutions.drawing_styles

def get_hands(seq_df):
    images = []
    all_hand_landmarks = []
    for seq_idx in range(len(seq_df)):
        x_hand = seq_df.iloc[seq_idx].filter(regex="x_right_hand.*").values
        y_hand = seq_df.iloc[seq_idx].filter(regex="y_right_hand.*").values
        z_hand = seq_df.iloc[seq_idx].filter(regex="z_right_hand.*").values

        right_hand_image = np.zeros((600, 600, 3))

        right_hand_landmarks = landmark_pb2.NormalizedLandmarkList()

        for x, y, z in zip(x_hand, y_hand, z_hand):
            right_hand_landmarks.landmark.add(x=x, y=y, z=z)

        mp_drawing.draw_landmarks(
                right_hand_image,
                right_hand_landmarks,
                mp_hands.HAND_CONNECTIONS,
                landmark_drawing_spec=mp_drawing_styles.get_default_hand_landmarks_sty

        x_hand = seq_df.iloc[seq_idx].filter(regex="x_left_hand.*").values
        y_hand = seq_df.iloc[seq_idx].filter(regex="y_left_hand.*").values
        z_hand = seq_df.iloc[seq_idx].filter(regex="z_left_hand.*").values

        left_hand_image = np.zeros((600, 600, 3))

        left_hand_landmarks = landmark_pb2.NormalizedLandmarkList()
        for x, y, z in zip(x_hand, y_hand, z_hand):
            left_hand_landmarks.landmark.add(x=x, y=y, z=z)

        mp_drawing.draw_landmarks(
                left_hand_image,
                left_hand_landmarks,
```

```
                mp_hands.HAND_CONNECTIONS,
                landmark_drawing_spec=mp_drawing_styles.get_default_hand_landmarks_sty

        images.append([right_hand_image.astype(np.uint8), left_hand_image.astype(np.ui
        all_hand_landmarks.append([right_hand_landmarks, left_hand_landmarks])
    return images, all_hand_landmarks
```

In [11]:

```
# Get the images created using mediapipe apis
hand_images, hand_landmarks = get_hands(sample_sequence_df)
# Fetch and show the data for right hand
create_animation(np.array(hand_images)[:, 0])
```

Out[11]:



Once Loop Reflect

○ Once ◉ Loop ○ Reflect

Preprocess the data

"For the sake of convenience and efficiency, we will reorganize the data so that each Parquet file contains both the landmark data and the corresponding phrase, eliminating the need to switch between train.csv and its Parquet file.

The new data will be saved in the [TFRecord] format. TFRecord is a streamlined format for storing binary records, which offers significantly improved data storage and retrieval performance."

ASL fingerspelling primarily focuses on hand movements. Therefore, we will use hand landmark coordinates and hand pose coordinates to train the model.

Fetch the pose landmark coordinates related to hand movement.

In [12]:
```
# Pose coordinates for hand movement.
LPOSE = [13, 15, 17, 19, 21]
RPOSE = [14, 16, 18, 20, 22]
POSE = LPOSE + RPOSE
```

Create x,y,z label names from coordinates

In [13]:
```
X = [f'x_right_hand_{i}' for i in range(21)] + [f'x_left_hand_{i}' for i in range(21)]
Y = [f'y_right_hand_{i}' for i in range(21)] + [f'y_left_hand_{i}' for i in range(21)]
Z = [f'z_right_hand_{i}' for i in range(21)] + [f'z_left_hand_{i}' for i in range(21)]
```

Create feature columns from the extracted coordinates.

In [14]:
```
FEATURE_COLUMNS = X + Y + Z
```

Store ids of each coordinate labels to lists

In [15]:
```
X_IDX = [i for i, col in enumerate(FEATURE_COLUMNS)  if "x_" in col]
Y_IDX = [i for i, col in enumerate(FEATURE_COLUMNS)  if "y_" in col]
Z_IDX = [i for i, col in enumerate(FEATURE_COLUMNS)  if "z_" in col]

RHAND_IDX = [i for i, col in enumerate(FEATURE_COLUMNS)  if "right" in col]
LHAND_IDX = [i for i, col in enumerate(FEATURE_COLUMNS)  if  "left" in col]
RPOSE_IDX = [i for i, col in enumerate(FEATURE_COLUMNS)  if  "pose" in col and int(col
LPOSE_IDX = [i for i, col in enumerate(FEATURE_COLUMNS)  if  "pose" in col and int(col
```

Preprocess and write the dataset as TFRecords

Using the extracted landmarks and phrases, we will create new dataset files and save them as TFRecords.

In [16]:
```
# Set length of frames to 128
FRAME_LEN = 128

# Create directory to store the new data
if not os.path.isdir("preprocessed"):
    os.mkdir("preprocessed")
else:
    shutil.rmtree("preprocessed")
    os.mkdir("preprocessed")

# Loop through each file_id
for file_id in tqdm(dataset_df.file_id.unique()):
    # Parquet file name
    pq_file = f"/input/asl-fingerspelling/train_landmarks/{file_id}.parquet"
    # Filter train.csv and fetch entries only for the relevant file_id
    file_df = dataset_df.loc[dataset_df["file_id"] == file_id]
    # Fetch the parquet file
    parquet_df = pq.read_table(f"/input/asl-fingerspelling/train_landmarks/{str(file_i
                               columns=['sequence_id'] + FEATURE_COLUMNS).to_pandas()
    # File name for the updated data
    tf_file = f"preprocessed/{file_id}.tfrecord"
    parquet_numpy = parquet_df.to_numpy()
    # Initialize the pointer to write the output of
    # each `for loop` below as a sequence into the file.
    with tf.io.TFRecordWriter(tf_file) as file_writer:
        # Loop through each sequence in file.
        for seq_id, phrase in zip(file_df.sequence_id, file_df.phrase):
            # Fetch sequence data
            frames = parquet_numpy[parquet_df.index == seq_id]

            # Calculate the number of NaN values in each hand landmark
            r_nonan = np.sum(np.sum(np.isnan(frames[:, RHAND_IDX]), axis = 1) == 0)
            l_nonan = np.sum(np.sum(np.isnan(frames[:, LHAND_IDX]), axis = 1) == 0)
            no_nan = max(r_nonan, l_nonan)

            if 2*len(phrase)<no_nan:
                features = {FEATURE_COLUMNS[i]: tf.train.Feature(
                    float_list=tf.train.FloatList(value=frames[:, i])) for i in range(
                features["phrase"] = tf.train.Feature(bytes_list=tf.train.BytesList(va
                record_bytes = tf.train.Example(features=tf.train.Features(feature=fea
                file_writer.write(record_bytes)
```

```
  0%|          | 0/68 [00:00<?, ?it/s]
```

Load the preprocessed data

Get the saved TFRecord files into a list

In [17]:
```
tf_records = dataset_df.file_id.map(lambda x: f'/working/preprocessed/{x}.tfrecord').u
print(f"List of {len(tf_records)} TFRecord files.")
```

```
List of 68 TFRecord files.
```

Load character_to_prediction json file

This json file contains a character and its value. We will add three new characters, "<" and ">" to mark the start and end of each phrase, and "P" for padding.

In [18]:
```
with open ("/input/asl-fingerspelling/character_to_prediction_index.json", "r") as f:
    char_to_num = json.load(f)

# Add pad_token, start pointer and end pointer to the dict
pad_token = 'P'
start_token = '<'
end_token = '>'
pad_token_idx = 59
start_token_idx = 60
end_token_idx = 61

char_to_num[pad_token] = pad_token_idx
char_to_num[start_token] = start_token_idx
char_to_num[end_token] = end_token_idx
num_to_char = {j:i for i,j in char_to_num.items()}
```

In [19]:
```
# Function to resize and add padding.
def resize_pad(x):
    if tf.shape(x)[0] < FRAME_LEN:
        x = tf.pad(x, ([[0, FRAME_LEN-tf.shape(x)[0]], [0, 0], [0, 0]]))
    else:
        x = tf.image.resize(x, (FRAME_LEN, tf.shape(x)[1]))
    return x

# Detect the dominant hand from the number of NaN values.
# Dominant hand will have less NaN values since it is in frame moving.
def pre_process(x):
    rhand = tf.gather(x, RHAND_IDX, axis=1)
    lhand = tf.gather(x, LHAND_IDX, axis=1)
    rpose = tf.gather(x, RPOSE_IDX, axis=1)
    lpose = tf.gather(x, LPOSE_IDX, axis=1)

    rnan_idx = tf.reduce_any(tf.math.is_nan(rhand), axis=1)
    lnan_idx = tf.reduce_any(tf.math.is_nan(lhand), axis=1)

    rnans = tf.math.count_nonzero(rnan_idx)
    lnans = tf.math.count_nonzero(lnan_idx)

    # For dominant hand
    if rnans > lnans:
        hand = lhand
        pose = lpose

        hand_x = hand[:, 0*(len(LHAND_IDX)//3) : 1*(len(LHAND_IDX)//3)]
        hand_y = hand[:, 1*(len(LHAND_IDX)//3) : 2*(len(LHAND_IDX)//3)]
        hand_z = hand[:, 2*(len(LHAND_IDX)//3) : 3*(len(LHAND_IDX)//3)]
        hand = tf.concat([1-hand_x, hand_y, hand_z], axis=1)

        pose_x = pose[:, 0*(len(LPOSE_IDX)//3) : 1*(len(LPOSE_IDX)//3)]
        pose_y = pose[:, 1*(len(LPOSE_IDX)//3) : 2*(len(LPOSE_IDX)//3)]
        pose_z = pose[:, 2*(len(LPOSE_IDX)//3) : 3*(len(LPOSE_IDX)//3)]
        pose = tf.concat([1-pose_x, pose_y, pose_z], axis=1)
    else:
        hand = rhand
        pose = rpose

    hand_x = hand[:, 0*(len(LHAND_IDX)//3) : 1*(len(LHAND_IDX)//3)]
    hand_y = hand[:, 1*(len(LHAND_IDX)//3) : 2*(len(LHAND_IDX)//3)]
```

```
    hand_z = hand[:, 2*(len(LHAND_IDX)//3) : 3*(len(LHAND_IDX)//3)]
    hand = tf.concat([hand_x[..., tf.newaxis], hand_y[..., tf.newaxis], hand_z[..., tf

    mean = tf.math.reduce_mean(hand, axis=1)[:, tf.newaxis, :]
    std = tf.math.reduce_std(hand, axis=1)[:, tf.newaxis, :]
    hand = (hand - mean) / std

    pose_x = pose[:, 0*(len(LPOSE_IDX)//3) : 1*(len(LPOSE_IDX)//3)]
    pose_y = pose[:, 1*(len(LPOSE_IDX)//3) : 2*(len(LPOSE_IDX)//3)]
    pose_z = pose[:, 2*(len(LPOSE_IDX)//3) : 3*(len(LPOSE_IDX)//3)]
    pose = tf.concat([pose_x[..., tf.newaxis], pose_y[..., tf.newaxis], pose_z[..., tf

    x = tf.concat([hand, pose], axis=1)
    x = resize_pad(x)

    x = tf.where(tf.math.is_nan(x), tf.zeros_like(x), x)
    x = tf.reshape(x, (FRAME_LEN, len(LHAND_IDX) + len(LPOSE_IDX)))
    return x
```

Create function to parse data from TFRecord format

This function will read the `TFRecord` data and convert it to Tensors.

In [20]:
```
def decode_fn(record_bytes):
    schema = {COL: tf.io.VarLenFeature(dtype=tf.float32) for COL in FEATURE_COLUMNS}
    schema["phrase"] = tf.io.FixedLenFeature([], dtype=tf.string)
    features = tf.io.parse_single_example(record_bytes, schema)
    phrase = features["phrase"]
    landmarks = ([tf.sparse.to_dense(features[COL]) for COL in FEATURE_COLUMNS])
    # Transpose to maintain the original shape of landmarks data.
    landmarks = tf.transpose(landmarks)

    return landmarks, phrase
```

Create function to convert the data

This function transposes and applies masks to the landmark coordinates. It also vectorizes the phrase corresponding to the landmarks using character_to_prediction_index.json.

In [21]:
```
table = tf.lookup.StaticHashTable(
    initializer=tf.lookup.KeyValueTensorInitializer(
        keys=list(char_to_num.keys()),
        values=list(char_to_num.values()),
    ),
    default_value=tf.constant(-1),
    name="class_weight"
)

def convert_fn(landmarks, phrase):
    # Add start and end pointers to phrase.
    phrase = start_token + phrase + end_token
    phrase = tf.strings.bytes_split(phrase)
    phrase = table.lookup(phrase)
    # Vectorize and add padding.
    phrase = tf.pad(phrase, paddings=[[0, 64 - tf.shape(phrase)[0]]], mode = 'CONSTANT
                    constant_values = pad_token_idx)
```

```
    # Apply pre_process function to the landmarks.
    return pre_process(landmarks), phrase
```

Use the functions we defined above to create the final dataset.

Train and validation dataset split

In [22]:
```
batch_size = 64
train_len = int(0.8 * len(tf_records))

train_ds = tf.data.TFRecordDataset(tf_records[:train_len]).map(decode_fn).map(convert_
valid_ds = tf.data.TFRecordDataset(tf_records[train_len:]).map(decode_fn).map(convert_
```

Create the Transformer model

Define the Transformer Input Layers

"When processing landmark coordinate features for the encoder, convolutional layers are applied to downsample and capture local relationships.

For the decoder, we combine position embeddings and token embeddings when processing past target tokens."

In [23]:
```
class TokenEmbedding(layers.Layer):
    def __init__(self, num_vocab=1000, maxlen=100, num_hid=64):
        super().__init__()
        self.emb = tf.keras.layers.Embedding(num_vocab, num_hid)
        self.pos_emb = layers.Embedding(input_dim=maxlen, output_dim=num_hid)

    def call(self, x):
        maxlen = tf.shape(x)[-1]
        x = self.emb(x)
        positions = tf.range(start=0, limit=maxlen, delta=1)
        positions = self.pos_emb(positions)
        return x + positions


class LandmarkEmbedding(layers.Layer):
    def __init__(self, num_hid=64, maxlen=100):
        super().__init__()
        self.conv1 = tf.keras.layers.Conv1D(
            num_hid, 11, strides=2, padding="same", activation="relu"
        )
        self.conv2 = tf.keras.layers.Conv1D(
            num_hid, 11, strides=2, padding="same", activation="relu"
        )
        self.conv3 = tf.keras.layers.Conv1D(
            num_hid, 11, strides=2, padding="same", activation="relu"
        )
        self.pos_emb = layers.Embedding(input_dim=maxlen, output_dim=num_hid)

    def call(self, x):
        x = self.conv1(x)
```

```
        x = self.conv2(x)
        return self.conv3(x)
```

Encoder layer for Transformer

In [24]:
```
class TransformerEncoder(layers.Layer):
    def __init__(self, embed_dim, num_heads, feed_forward_dim, rate=0.1):
        super().__init__()
        self.att = layers.MultiHeadAttention(num_heads=num_heads, key_dim=embed_dim)
        self.ffn = keras.Sequential(
            [
                layers.Dense(feed_forward_dim, activation="relu"),
                layers.Dense(embed_dim),
            ]
        )
        self.layernorm1 = layers.LayerNormalization(epsilon=1e-6)
        self.layernorm2 = layers.LayerNormalization(epsilon=1e-6)
        self.dropout1 = layers.Dropout(rate)
        self.dropout2 = layers.Dropout(rate)

    def call(self, inputs, training):
        attn_output = self.att(inputs, inputs)
        attn_output = self.dropout1(attn_output, training=training)
        out1 = self.layernorm1(inputs + attn_output)
        ffn_output = self.ffn(out1)
        ffn_output = self.dropout2(ffn_output, training=training)
        return self.layernorm2(out1 + ffn_output)
```

Decoder layer for Transformer

In [25]:
```
# Customized to add `training` variable
class TransformerDecoder(layers.Layer):
    def __init__(self, embed_dim, num_heads, feed_forward_dim, dropout_rate=0.1):
        super().__init__()
        self.layernorm1 = layers.LayerNormalization(epsilon=1e-6)
        self.layernorm2 = layers.LayerNormalization(epsilon=1e-6)
        self.layernorm3 = layers.LayerNormalization(epsilon=1e-6)
        self.self_att = layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=embed_dim
        )
        self.enc_att = layers.MultiHeadAttention(num_heads=num_heads, key_dim=embed_di
        self.self_dropout = layers.Dropout(0.5)
        self.enc_dropout = layers.Dropout(0.1)
        self.ffn_dropout = layers.Dropout(0.1)
        self.ffn = keras.Sequential(
            [
                layers.Dense(feed_forward_dim, activation="relu"),
                layers.Dense(embed_dim),
            ]
        )

    def causal_attention_mask(self, batch_size, n_dest, n_src, dtype):
        """Masks the upper half of the dot product matrix in self attention.

        This prevents flow of information from future tokens to current token.
        1's in the lower triangle, counting from the lower right corner.
        """
```

```
        i = tf.range(n_dest)[:, None]
        j = tf.range(n_src)
        m = i >= j - n_src + n_dest
        mask = tf.cast(m, dtype)
        mask = tf.reshape(mask, [1, n_dest, n_src])
        mult = tf.concat(
            [batch_size[..., tf.newaxis], tf.constant([1, 1], dtype=tf.int32)], 0
        )
        return tf.tile(mask, mult)

    def call(self, enc_out, target, training):
        input_shape = tf.shape(target)
        batch_size = input_shape[0]
        seq_len = input_shape[1]
        causal_mask = self.causal_attention_mask(batch_size, seq_len, seq_len, tf.bool
        target_att = self.self_att(target, target, attention_mask=causal_mask)
        target_norm = self.layernorm1(target + self.self_dropout(target_att, training
        enc_out = self.enc_att(target_norm, enc_out)
        enc_out_norm = self.layernorm2(self.enc_dropout(enc_out, training = training)
        ffn_out = self.ffn(enc_out_norm)
        ffn_out_norm = self.layernorm3(enc_out_norm + self.ffn_dropout(ffn_out, traini
        return ffn_out_norm
```

Complete the Transformer model

"This model takes landmark coordinates as inputs and predicts a sequence of characters. During training, the target character sequence, shifted to the left, serves as the input to the decoder. The decoder uses its past predictions during inference to forecast the next token.

The accuracy metric for this contest is the Levenshtein Distance between sequences, as the evaluation metric is the Normalized Total Levenshtein Distance."

In [26]:
```
# Customized to add edit_dist metric and training variable.

class Transformer(keras.Model):
    def __init__(
        self,
        num_hid=64,
        num_head=2,
        num_feed_forward=128,
        source_maxlen=100,
        target_maxlen=100,
        num_layers_enc=4,
        num_layers_dec=1,
        num_classes=60,
    ):
        super().__init__()
        self.loss_metric = keras.metrics.Mean(name="loss")
        self.acc_metric = keras.metrics.Mean(name="edit_dist")
        self.num_layers_enc = num_layers_enc
        self.num_layers_dec = num_layers_dec
        self.target_maxlen = target_maxlen
        self.num_classes = num_classes

        self.enc_input = LandmarkEmbedding(num_hid=num_hid, maxlen=source_maxlen)
        self.dec_input = TokenEmbedding(
            num_vocab=num_classes, maxlen=target_maxlen, num_hid=num_hid
```

```
        )

        self.encoder = keras.Sequential(
            [self.enc_input]
            + [
                TransformerEncoder(num_hid, num_head, num_feed_forward)
                for _ in range(num_layers_enc)
            ]
        )

        for i in range(num_layers_dec):
            setattr(
                self,
                f"dec_layer_{i}",
                TransformerDecoder(num_hid, num_head, num_feed_forward),
            )

        self.classifier = layers.Dense(num_classes)

    def decode(self, enc_out, target, training):
        y = self.dec_input(target)
        for i in range(self.num_layers_dec):
            y = getattr(self, f"dec_layer_{i}")(enc_out, y, training)
        return y

    def call(self, inputs, training):
        source = inputs[0]
        target = inputs[1]
        x = self.encoder(source, training)
        y = self.decode(x, target, training)
        return self.classifier(y)

    @property
    def metrics(self):
        return [self.loss_metric]

    def train_step(self, batch):
        """Processes one batch inside model.fit()."""
        source = batch[0]
        target = batch[1]

        input_shape = tf.shape(target)
        batch_size = input_shape[0]

        dec_input = target[:, :-1]
        dec_target = target[:, 1:]
        with tf.GradientTape() as tape:
            preds = self([source, dec_input])
            one_hot = tf.one_hot(dec_target, depth=self.num_classes)
            mask = tf.math.logical_not(tf.math.equal(dec_target, pad_token_idx))
            loss = self.compiled_loss(one_hot, preds, sample_weight=mask)
        trainable_vars = self.trainable_variables
        gradients = tape.gradient(loss, trainable_vars)
        self.optimizer.apply_gradients(zip(gradients, trainable_vars))
        # Computes the Levenshtein distance between sequences since the evaluation
        # metric for this contest is the normalized total levenshtein distance.
        edit_dist = tf.edit_distance(tf.sparse.from_dense(target),
                                     tf.sparse.from_dense(tf.cast(tf.argmax(preds, axi
        edit_dist = tf.reduce_mean(edit_dist)
        self.acc_metric.update_state(edit_dist)
```

```
        self.loss_metric.update_state(loss)
        return {"loss": self.loss_metric.result(), "edit_dist": self.acc_metric.result

    def test_step(self, batch):
        source = batch[0]
        target = batch[1]

        input_shape = tf.shape(target)
        batch_size = input_shape[0]

        dec_input = target[:, :-1]
        dec_target = target[:, 1:]
        preds = self([source, dec_input])
        one_hot = tf.one_hot(dec_target, depth=self.num_classes)
        mask = tf.math.logical_not(tf.math.equal(dec_target, pad_token_idx))
        loss = self.compiled_loss(one_hot, preds, sample_weight=mask)
        # Computes the Levenshtein distance between sequences since the evaluation
        # metric for this contest is the normalized total levenshtein distance.
        edit_dist = tf.edit_distance(tf.sparse.from_dense(target),
                                     tf.sparse.from_dense(tf.cast(tf.argmax(preds, axi
        edit_dist = tf.reduce_mean(edit_dist)
        self.acc_metric.update_state(edit_dist)
        self.loss_metric.update_state(loss)
        return {"loss": self.loss_metric.result(), "edit_dist": self.acc_metric.result

    def generate(self, source, target_start_token_idx):
        """Performs inference over one batch of inputs using greedy decoding."""
        bs = tf.shape(source)[0]
        enc = self.encoder(source, training = False)
        dec_input = tf.ones((bs, 1), dtype=tf.int32) * target_start_token_idx
        dec_logits = []
        for i in range(self.target_maxlen - 1):
            dec_out = self.decode(enc, dec_input, training = False)
            logits = self.classifier(dec_out)
            logits = tf.argmax(logits, axis=-1, output_type=tf.int32)
            last_logit = logits[:, -1][..., tf.newaxis]
            dec_logits.append(last_logit)
            dec_input = tf.concat([dec_input, last_logit], axis=-1)
        return dec_input
```

The following callback function is used to display predictions.

In [27]:
```
class DisplayOutputs(keras.callbacks.Callback):
    def __init__(
        self, batch, idx_to_token, target_start_token_idx=60, target_end_token_idx=61
    ):
        """Displays a batch of outputs after every 4 epoch

        Args:
            batch: A test batch
            idx_to_token: A List containing the vocabulary tokens corresponding to the
            target_start_token_idx: A start token index in the target vocabulary
            target_end_token_idx: An end token index in the target vocabulary
        """
        self.batch = batch
        self.target_start_token_idx = target_start_token_idx
        self.target_end_token_idx = target_end_token_idx
        self.idx_to_char = idx_to_token
```

```
    def on_epoch_end(self, epoch, logs=None):
        if epoch % 4 != 0:
            return
        source = self.batch[0]
        target = self.batch[1].numpy()
        bs = tf.shape(source)[0]
        preds = self.model.generate(source, self.target_start_token_idx)
        preds = preds.numpy()
        for i in range(bs):
            target_text = "".join([self.idx_to_char[_] for _ in target[i, :]])
            prediction = ""
            for idx in preds[i, :]:
                prediction += self.idx_to_char[idx]
                if idx == self.target_end_token_idx:
                    break
            print(f"target:     {target_text.replace('-','')}")
            print(f"prediction: {prediction}\n")
```

Train the Transformer model

In [28]:

```
# Transformer variables are customized from original keras tutorial to suit this datas

batch = next(iter(valid_ds))

# The vocabulary to convert predicted indices into characters
idx_to_char = list(char_to_num.keys())
display_cb = DisplayOutputs(
    batch, idx_to_char, target_start_token_idx=char_to_num['<'], target_end_token_idx=
)  # set the arguments as per vocabulary index for '<' and '>'

model = Transformer(
    num_hid=200,
    num_head=4,
    num_feed_forward=400,
    source_maxlen = FRAME_LEN,
    target_maxlen=64,
    num_layers_enc=2,
    num_layers_dec=1,
    num_classes=62
)
loss_fn = tf.keras.losses.CategoricalCrossentropy(
    from_logits=True, label_smoothing=0.1,
)


optimizer = keras.optimizers.Adam(0.0001)
model.compile(optimizer=optimizer, loss=loss_fn)

history = model.fit(train_ds, validation_data=valid_ds, callbacks=[display_cb], epochs
```

```
Epoch 1/13
    632/Unknown - 69s 76ms/step - loss: 0.8790 - edit_dist: 1.0449target:     <2796 w
est golden willow drive>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <785 stest wererere rere>

target:     <9734719887>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+69-261-2-29>

target:     <4977236992>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <989-166-2662>

target:     <reallyloud.co.uk/simaii>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <derene derestes>

target:     <kkaicd1.pixnet.net>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <https://www.comangananaranas>

target:     <8260 john r bowdoin>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <868 ware weresth re>

target:     <56 paper birch drive>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <406-201-7717>

target:     <gandchudaihardcor.html>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <2039 hangh roronge>

target:     <2708 west 77th>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <186-661-171>

target:     <https://www.keainfo.gr>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <antps://www.comaricom>

target:     <288 fuller lake>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <288-981-5113>

target:     <mser/okiguide>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <serin rinen>

target:     <220 north 47th avenue east>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <322 hanth panest>

target:     <www.sudinfo.be>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.nerarere>

target:     <69 grant point>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <639 ste rere>

target:     <fibrain.pl/chapter145/thanghr>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <990-11-2-111>

target:     <+3515218895>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+33-393-3858>

target:     <viapierogobetti/brandilove>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <222 siesthin comine>

target:     <amir le>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <anen serere>

target:     <automantenimientosa>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
```

```
prediction: <manestestestestes>

target:     <6499 nfd 5053>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <695-859-5723>

target:     <7870 preston place>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <391-181-185-5>

target:     <6870 scabisuit lane>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <682 sianin st rin>

target:     <sunshine mayer>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <anthan sthaneran>

target:     <4082494707>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <482-253-7-20>

target:     <arabradio.us/vana>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <anare stareranest>

target:     <televisoresponse>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <6800 core re rand>

target:     <www.voices.com/bitesize>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <626-060-0686>

target:     <qd.razavi.ac.ir/bunya>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: </rerananarestererandeste>

target:     <+246987508002294>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+66-868-8-9-9-949-5>

target:     <+2666480514>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <226-800-8808>

target:     <2257645994>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <235-559-5949>

target:     <35816 webwood>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <380-060-0000>

target:     <liana deleon>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <anan sananen>

target:     </p21sistemas/sharps_carbines>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <https://ww.com/mareresereste>

target:     <www.boehringeringelheim.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.haringeringengend>

target:     <2901092582>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <3565 st ste st rtestet>

target:     <102 flagstad>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <190 stan sthan>

target:     </verkstader/20180101>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <330-020-0010>

target:     <myanmarsocialonlinemedia.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
```

```
prediction: <anere stestestes>

target:     <bunkerbranch.tumblr.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <692 sthan wanthereroure>

target:     <+8286698863353>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+88-686-6-23>

target:     <9464 east woods edge way>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.stanestesthesterine>

target:     <4632526021>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <462-262-6222>

target:     <martin+blando/>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <https://ww.com/>

target:     <beloshveyka.ru/105>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <900-101-2018>

target:     <bid.auctionbymayo.com/71063>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <meresteste berararad>

target:     <7384391127>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <353-553-5631>

target:     <www.twitchquotes.com/>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.stanestines.com/>

target:     <5016 east paquin street>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <461 stant st st sterast>

target:     <2292979260>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <243-266-6666>

target:     <9186054830>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <911-818-1612>

target:     <brandee tran>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <300 rore rere>

target:     <5061 williamsburg drive east>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <beristine st rananananas>

target:     <certuspro.com/bloqueosjalapa>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <berestestestestestest>

target:     <www.tudecora.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.corestereres.com/>

target:     <+279053128>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+39-923-91-5>

target:     <rachael salas>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <http:///www.comaneste>

target:     <9525366907>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <936-666-6905>

target:     <963 crested owl>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
```

```
prediction: <992 stort we rorte rt>

target:     <2886 will teasley>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <2886 st walane rad>

target:     <josie savage>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: </arestesthestestesth part>

target:     <4127407457>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <4474 s sth roart>

target:     <www.tornadoshop.cz/cornelius>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.tortondest.com/>

632/632 [==============================] - 83s 98ms/step - loss: 0.8790 - edit_dist:
1.0449 - val_loss: 0.8054 - val_edit_dist: 1.0455
Epoch 2/13
632/632 [==============================] - 21s 34ms/step - loss: 0.7538 - edit_dist:
1.0465 - val_loss: 0.7216 - val_edit_dist: 1.0469
Epoch 3/13
632/632 [==============================] - 21s 33ms/step - loss: 0.6789 - edit_dist:
1.0465 - val_loss: 0.6638 - val_edit_dist: 1.0458
Epoch 4/13
632/632 [==============================] - 21s 33ms/step - loss: 0.6304 - edit_dist:
1.0452 - val_loss: 0.6344 - val_edit_dist: 1.0443
Epoch 5/13
631/632 [============================>.] - ETA: 0s - loss: 0.5968 - edit_dist: 1.0437
target:     <2796 west golden willow drive>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <7796 west wilest d drive>

target:     <9734719887>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+36-31-41-19>

target:     <4977236992>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <498-823-6692>

target:     <reallyloud.co.uk/simaii>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <relldu/simaiomai>

target:     <kkaicd1.pixnet.net>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <hablida-p-pe-ten-e-tetetetes>

target:     <8260 john r bowdoin>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <826 mihanubew wdrd>

target:     <56 paper birch drive>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <56 neabl rch>

target:     <gandchudaihardcor.html>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <roshetipar.co/>

target:     <2708 west 77th>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <7286 westh 770>

target:     <https://www.keainfo.gr>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <https://www.ainf.fom>

target:     <288 fuller lake>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <288-919-6131>
```

```
target:     <mser/okiguide>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <serkit guide>

target:     <220 north 47th avenue east>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <220 north an has>

target:     <www.sudinfo.be>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.skinff.com>

target:     <69 grant point>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <629 wandis>

target:     <fibrain.pl/chapter145/thanghr>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <firaia-11110.bl/kanganur>

target:     <+3515218895>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+355-555-111-89>

target:     <viapierogobetti/brandilove>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <220 piretie th wadilive>

target:     <amir le>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <amie le>

target:     <automantenimientosa>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <manientoie miet>

target:     <6499 nfd 5053>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <649-918-5015>

target:     <7870 preston place>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <952-801-505-505>

target:     <6870 scabisuit lane>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <6770 sas sarit lave>

target:     <sunshine mayer>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <sutashin maler>

target:     <4082494707>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <488-834-4500>

target:     <arabradio.us/vana>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <araba diolers/ves/va>

target:     <televisoresponse>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <160 we fastor rt>

target:     <www.voices.com/bitesize>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.vece.com/bositisssit>

target:     <qd.razavi.ac.ir/bunya>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: </olera-gacelitas/>

target:     <+246987508002294>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+246-988-80-0294>

target:     <+2666480514>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+224-64-80-514-514>
```

```
target:     <2257645994>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <255-576-5941>

target:     <35816 webwood>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <388 cow bod>

target:     <liana deleon>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <allia weretas>

target:     </p21sistemas/sharps_carbines>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <glesis-e-marabarie>

target:     <www.boehringeringelheim.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.cechingerterte peget.com>

target:     <2901092582>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <290-109-1042>

target:     <102 flagstad>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <1029 aget lal>

target:     </verkstader/20180101>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <2 rustut 2080t 200>

target:     <myanmarsocialonlinemedia.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <anandocaraleemi.com>

target:     <bunkerbranch.tumblr.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <bundbanch.re/tuncom/>

target:     <+8286698863353>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+82-86-86-86-233>

target:     <9464 east woods edge way>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.w.w.w.w.com/>

target:     <4632526021>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <463-252-6021>

target:     <martin+blando/>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <https://lelonge.com/>

target:     <beloshveyka.ru/105>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <belot-sueveru/>

target:     <bid.auctionbymayo.com/71063>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <ebitbl.coba/bateba-do-201013>

target:     <7384391127>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <735-439-121-125>

target:     <www.twitchquotes.com/>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.twitate.com/>

target:     <5016 east paquin street>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <5501 weast pavet>

target:     <2292979260>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <299-297-2666>
```

```
target:     <9186054830>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <918-883-4833>

target:     <brandee tran>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <buande lun>

target:     <5061 williamsburg drive east>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <oww.willismer bur driveast>

target:     <certuspro.com/bloqueosjalapa>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <coutus-dopo.com/blom/bla>

target:     <www.tudecora.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.vedece/trarara.com>

target:     <+279053128>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+27-9000-124>

target:     <rachael salas>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <https://shales>

target:     <9525366907>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <982-526-6907>

target:     <963 crested owl>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <963 rost 106 rowad>

target:     <2886 will teasley>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <286 wiltea lle>

target:     <josie savage>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <hostes steagling>

target:     <4127407457>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <451-603-7744>

target:     <www.tornadoshop.cz/cornelius>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.ondosh.do/corkilikikes>

632/632 [==============================] - 23s 36ms/step - loss: 0.5966 - edit_dist:
1.0437 - val_loss: 0.6140 - val_edit_dist: 1.0429
Epoch 6/13
632/632 [==============================] - 21s 33ms/step - loss: 0.5693 - edit_dist:
1.0424 - val_loss: 0.5935 - val_edit_dist: 1.0417
Epoch 7/13
632/632 [==============================] - 21s 33ms/step - loss: 0.5465 - edit_dist:
1.0413 - val_loss: 0.5780 - val_edit_dist: 1.0407
Epoch 8/13
632/632 [==============================] - 21s 33ms/step - loss: 0.5283 - edit_dist:
1.0403 - val_loss: 0.5671 - val_edit_dist: 1.0398
Epoch 9/13
631/632 [============================>.] - ETA: 0s - loss: 0.5113 - edit_dist: 1.0394
target:     <2796 west golden willow drive>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <1779 west lonwill delow drive>

target:     <9734719887>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+61-31-51-19>

target:     <4977236992>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <494-823-6921>
```

```
target:     <reallyloud.co.uk/simaii>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <realidud.com.ru/simai>

target:     <kkaicd1.pixnet.net>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <abanabd-pinet-tetet>

target:     <8260 john r bowdoin>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <826 mishnr bow droin>

target:     <56 paper birch drive>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <546 pens bir rd>

target:     <gandchudaihardcor.html>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <randia hubercert crouthad>

target:     <2708 west 77th>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <2786 west 77>

target:     <https://www.keainfo.gr>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <https://www.ainfo.jp>

target:     <288 fuller lake>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <288-921-0131>

target:     <mser/okiguide>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <seser/di-guide>

target:     <220 north 47th avenue east>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <220 north and hoas>

target:     <www.sudinfo.be>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.skinff.com>

target:     <69 grant point>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <629 wand mine>

target:     <fibrain.pl/chapter145/thanghr>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <firait-1110 roath drou>

target:     <+3515218895>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+355-255-355-11-89>

target:     <viapierogobetti/brandilove>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <2020 pirtie roadi/mandite>

target:     <amir le>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <amir lee>

target:     <automantenimientosa>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <manien miento-s>

target:     <6499 nfd 5053>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <649-901-5050>

target:     <7870 preston place>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <950 preen pl place>

target:     <6870 scabisuit lane>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <670 scabis scaritlan>
```

```
target:     <sunshine mayer>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <sutun shine maler>

target:     <4082494707>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+46-87-48-48-00>

target:     <arabradio.us/vana>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <aarabrasdios/vasers/vana>

target:     <televisoresponse>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <201 werest borada>

target:     <www.voices.com/bitesize>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.vopete.com/baresiti>

target:     <qd.razavi.ac.ir/bunya>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: </delor-ace-tacir-arura>

target:     <+246987508002294>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+246-98-80-802-2294>

target:     <+2666480514>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+222-64-80-80-514>

target:     <2257645994>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+225-764-59-94-44>

target:     <35816 webwood>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <38 rob bood>

target:     <liana deleon>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <allia katurest>

target:     </p21sistemas/sharps_carbines>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: </111 sistema/carbis>

target:     <www.boehringeringelheim.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.boechringertert dexhim>

target:     <2901092582>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <290-119-109-582>

target:     <102 flagstad>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <1029 agen aral>

target:     </verkstader/20180101>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.rusader208.com.oz>

target:     <myanmarsocialonlinemedia.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <manars socalleenda.com>

target:     <bunkerbranch.tumblr.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <bunder banchtatumbur.com>

target:     <+8286698863353>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+82-86-86-86-863-253>

target:     <9464 east woods edge way>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: </hastww.ww.tm.com/>
```

```
target:     <4632526021>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <463-252-6021>

target:     <martin+blando/>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <anadelande.com/>

target:     <beloshveyka.ru/105>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <belot-suerever/100105>

target:     <bid.auctionbymayo.com/71063>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <eebid-ridaribebatob.com/>

target:     <7384391127>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <735-439-1250>

target:     <www.twitchquotes.com/>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.twittwi.chom/>

target:     <5016 east paquin street>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <5501 weast pavin street>

target:     <2292979260>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <291-297-2916>

target:     <9186054830>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <918-148-056-05-483>

target:     <brandee tran>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <brande detran>

target:     <5061 williamsburg drive east>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <cow.williamsbur drive east>

target:     <certuspro.com/bloqueosjalapa>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <cobusporspo.com/blos-ia>

target:     <www.tudecora.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.bureectomecar.com>

target:     <+279053128>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+27-900-90-128>

target:     <rachael salas>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <karh melle sean>

target:     <9525366907>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <942-526-6907>

target:     <963 crested owl>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <963 rosted 100 wal>

target:     <2886 will teasley>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <2886 wiltea ldr>

target:     <josie savage>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <josestesenestug.f>

target:     <4127407457>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <451-603-7744>
```

```
target:     <www.tornadoshop.cz/cornelius>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.torndsse.co.com/corkerkis>

632/632 [==============================] - 23s 36ms/step - loss: 0.5111 - edit_dist:
1.0394 - val_loss: 0.5580 - val_edit_dist: 1.0390
Epoch 10/13
632/632 [==============================] - 21s 33ms/step - loss: 0.4962 - edit_dist:
1.0387 - val_loss: 0.5503 - val_edit_dist: 1.0383
Epoch 11/13
632/632 [==============================] - 21s 33ms/step - loss: 0.4841 - edit_dist:
1.0380 - val_loss: 0.5491 - val_edit_dist: 1.0377
Epoch 12/13
632/632 [==============================] - 21s 33ms/step - loss: 0.4749 - edit_dist:
1.0375 - val_loss: 0.5459 - val_edit_dist: 1.0372
Epoch 13/13
631/632 [============================>.] - ETA: 0s - loss: 0.4665 - edit_dist: 1.0370
target:     <2796 west golden willow drive>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <1795 west lonwillow drive>

target:     <9734719887>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+66-41-41-139>

target:     <4977236992>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <498-872-6923>

target:     <reallyloud.co.uk/simaii>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <realyud.com.u/simaimai>

target:     <kkaicd1.pixnet.net>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <abarabl-pistignet.net>

target:     <8260 john r bowdoin>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <8260 johnr bow din>

target:     <56 paper birch drive>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <56 pens brcherch drive>

target:     <gandchudaihardcor.html>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <aushory.com/dectorbechar>

target:     <2708 west 77th>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <2786 west 77>

target:     <https://www.keainfo.gr>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <https://www.preainfo.fox/>

target:     <288 fuller lake>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <2888 forll coul>

target:     <mser/okiguide>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <seser/oki-gride>

target:     <220 north 47th avenue east>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <20 north gread rive roas>

target:     <www.sudinfo.be>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.rukinffs.com>

target:     <69 grant point>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
```

```
prediction: <69 wan coine>

target:     <fibrain.pl/chapter145/thanghr>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <firain-11001/tertha-blander>

target:     <+3515218895>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+355-352-355-11-89>

target:     <viapierogobetti/brandilove>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <270 manir roagen drive>

target:     <amir le>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <amir lee>

target:     <automantenimientosa>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <marenieniento-sa>

target:     <6499 nfd 5053>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <649-909-0505>

target:     <7870 preston place>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <9350 preen place>

target:     <6870 scabisuit lane>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <670 scabis scarit lan>

target:     <sunshine mayer>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <sunshin maliner>

target:     <4082494707>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <478-547-4800>

target:     <arabradio.us/vana>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <aarabbrasdiosers/via>

target:     <televisoresponse>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <200 wooborfst woort da>

target:     <www.voices.com/bitesize>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.voroes.com/bie-staci>

target:     <qd.razavi.ac.ir/bunya>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: </olelocor-acire-gara>

target:     <+246987508002294>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+246-987-802-02-294>

target:     <+2666480514>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <224-64-801414>

target:     <2257645994>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+225-764-559-94>

target:     <35816 webwood>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <38 row drid>

target:     <liana deleon>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <allia kaduberos>

target:     </p21sistemas/sharps_carbines>PPPPPPPPPPPPPPPPPPPPPPPPPPPP
```

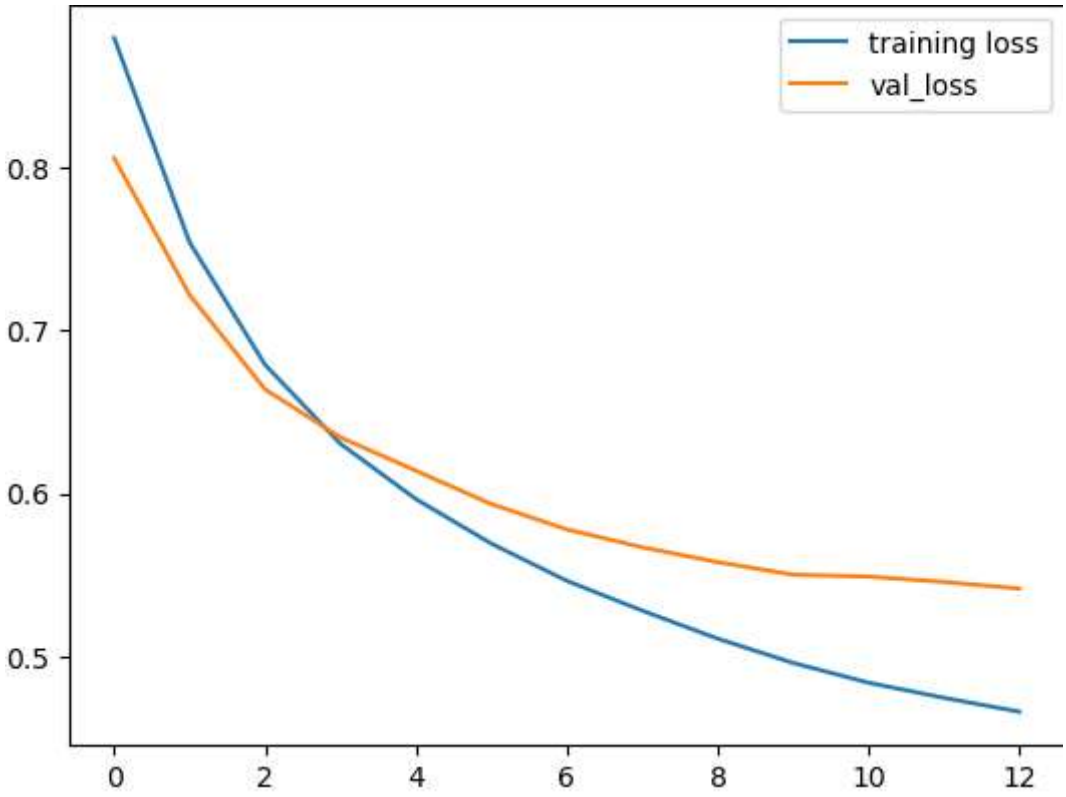
```
prediction: </lleis-estem/sharbarie>

target:     <www.boehringeringelheim.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.boechringer.ingelehom.com>

target:     <2901092582>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <290-1019-582>

target:     <102 flagstad>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <1029 louth last>

target:     </verkstader/20180101>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.rusader208.orgo.oz>

target:     <myanmarsocialonlinemedia.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <jansarsodgeorodeek.com>

target:     <bunkerbranch.tumblr.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <bunderbanch.rm/busanchom>

target:     <+8286698863353>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+82-86-86-863-253-253>

target:     <9464 east woods edge way>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: </307-wortw.com/ssancham>

target:     <4632526021>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <463-252-6021>

target:     <martin+blando/>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <ananie plroo/>

target:     <beloshveyka.ru/105>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <belot-shver/1000/1005>

target:     <bid.auctionbymayo.com/71063>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <eerbid-gribagribeb.com/>

target:     <7384391127>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <738-439-1270>

target:     <www.twitchquotes.com/>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.twitechontes.com/>

target:     <5016 east paquin street>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <5016 east pavin street>

target:     <2292979260>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <291-297-7516>

target:     <9186054830>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <918-485-0670>

target:     <brandee tran>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <brande delarkin>

target:     <5061 williamsburg drive east>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <306 williamsbur drive drive>

target:     <certuspro.com/bloqueosjalapa>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
```

```
prediction: <cobutuspo.com/4100s-ila>

target:     <www.tudecora.com>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.bardecarde/traric.com>

target:     <+279053128>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <+27-9053-128>

target:     <rachael salas>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <karhie mardes>

target:     <9525366907>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <942-576-6907>

target:     <963 crested owl>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <963 rirested owll>

target:     <2886 will teasley>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <2886 will teall>

target:     <josie savage>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <josessies/stuing-go>

target:     <4127407457>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <451-206-7744>

target:     <www.tornadoshop.cz/cornelius>PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
prediction: <www.torndosh.cz/counlirs>

632/632 [==============================] - 23s 36ms/step - loss: 0.4663 - edit_dist:
1.0370 - val_loss: 0.5418 - val_edit_dist: 1.0368
```

Plot training loss and validation loss

In [29]:
```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['training loss', 'val_loss'])
```

Out[29]:
```
<matplotlib.legend.Legend at 0x7fba6fc57100>
```



Create TFLite model

In [30]:
```
class TFLiteModel(tf.Module):
    def __init__(self, model):
        super(TFLiteModel, self).__init__()
        self.target_start_token_idx = start_token_idx
        self.target_end_token_idx = end_token_idx
        # Load the feature generation and main models
        self.model = model

    @tf.function(input_signature=[tf.TensorSpec(shape=[None, len(FEATURE_COLUMNS)], dt
    def __call__(self, inputs, training=False):
        # Preprocess Data
        x = tf.cast(inputs, tf.float32)
        x = x[None]
        x = tf.cond(tf.shape(x)[1] == 0, lambda: tf.zeros((1, 1, len(FEATURE_COLUMNS))
        x = x[0]
        x = pre_process(x)
        x = x[None]
        x = self.model.generate(x, self.target_start_token_idx)
        x = x[0]
        idx = tf.argmax(tf.cast(tf.equal(x, self.target_end_token_idx), tf.int32))
        idx = tf.where(tf.math.less(idx, 1), tf.constant(2, dtype=tf.int64), idx)
        x = x[1:idx]
        x = tf.one_hot(x, 59)
        return {'outputs': x}

tflitemodel_base = TFLiteModel(model)
```

In [31]:
```
model.save_weights("model.h5")
```

In [32]:
```
keras_model_converter = tf.lite.TFLiteConverter.from_keras_model(tflitemodel_base)
keras_model_converter.target_spec.supported_ops = [tf.lite.OpsSet.TFLITE_BUILTINS]#, t
```

```
tflite_model = keras_model_converter.convert()
with open('/working/model.tflite', 'wb') as f:
    f.write(tflite_model)

infargs = {"selected_columns" : FEATURE_COLUMNS}

with open('inference_args.json', "w") as json_file:
    json.dump(infargs, json_file)
```

In [33]:
```
!zip submission.zip  './model.tflite' './inference_args.json'
```

```
  adding: model.tflite (deflated 18%)
  adding: inference_args.json (deflated 85%)
```

In [35]:
```
interpreter = tf.lite.Interpreter("model.tflite")

REQUIRED_SIGNATURE = "serving_default"
REQUIRED_OUTPUT = "outputs"

with open ("/input/asl-fingerspelling/character_to_prediction_index.json", "r") as f:
    character_map = json.load(f)
rev_character_map = {j:i for i,j in character_map.items()}

found_signatures = list(interpreter.get_signature_list().keys())

if REQUIRED_SIGNATURE not in found_signatures:
    raise KernelEvalException('Required input signature not found.')

prediction_fn = interpreter.get_signature_runner("serving_default")
output = prediction_fn(inputs=batch[0][0])
prediction_str = "".join([rev_character_map.get(s, "") for s in np.argmax(output[REQUI
print(prediction_str)
```

```
885 ban
```