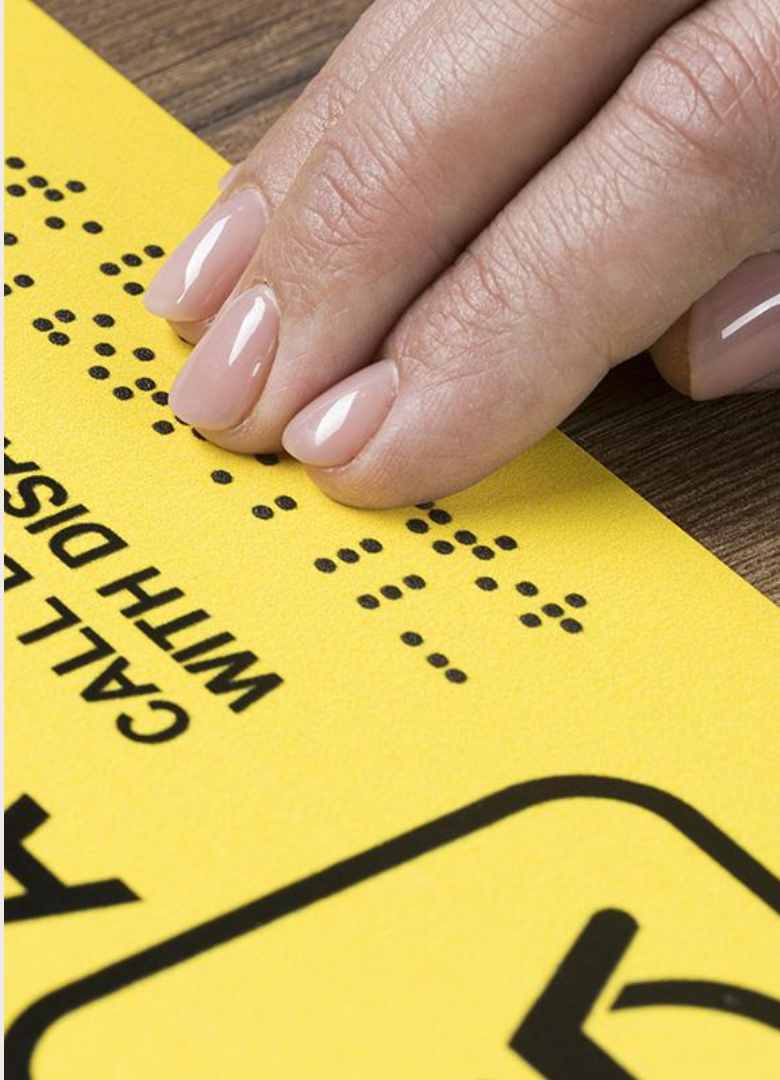
Braille Translation

Christina Carr & Mariam Germanyan

Introduction

The overall goal of our project is to take images in braille as input and output the english translation.

== ‘the’



Problem

Braille is widely used by people with visual impairments to read text. However, It isn’t a very efficient way of communicating information.

Solution

Image translation to help save time, increase efficiency when reading or translating Braille documents

Braille alphabet

A	B	C	D	E	F	G	H	I
								

J	K	L	M	N	O	P	Q	R
								

S	T	U	V	W	X	Y	Z
							

Keras Model

Model that predicts the braille image at the character level and returns the highest probability letter.

```
keras_model(braille_images[0]), english_words[0]
```

```
('hjp', 'hyp')
```

Language Model

Takes into account counts of uni-, bi-, tri-, and quad-character combinations

```
language_model('the')
```

```
[0.07654280007513439,
 0.311664926075615,
 0.6876460379116297,
 0.7650915050532642]
```

Combining the Models

Step 1:

Keras model outputs probabilities for each letter

Step 2:

Language model outputs probabilities for each word outputted by Keras

Step 3:

Multiply the output probabilities together

Step 4:

Return the word with the highest combined probability

Keras output word: “hjp”

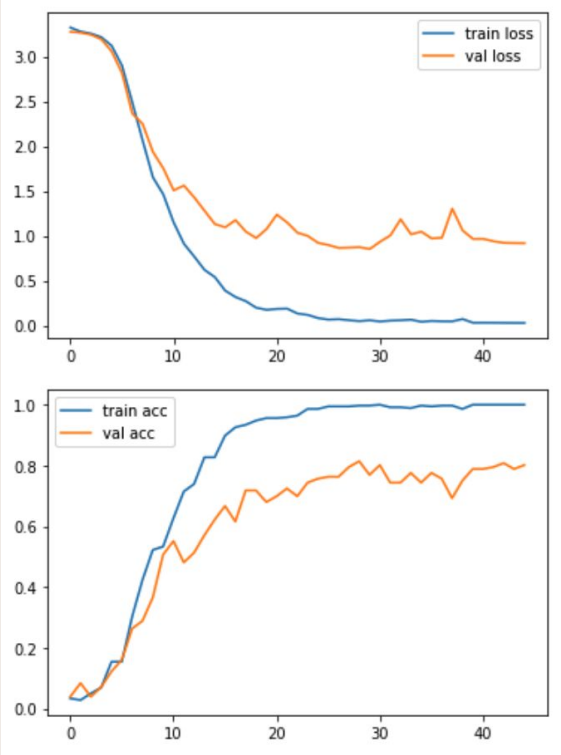
$$p_k(h) * p_{lm}(h)$$

$$\times p_k(j) * p_{lm}(h|j)$$

$$\times p_k(p) * p_{lm}(p|hj)$$

$$\times p_{lm}(/w|hjp)$$

Output




```
random_indices = np.random.choice(range(0, len(braille_images)), size = 10)
sample_actual = [english_words[i] for i in random_indices]
sample = [braille_images[i] for i in random_indices]
sample_output = [model_function(word, 5) for word in sample]
print('Predicted', sample_output)
print('Actual',sample_actual)
```

```
Predicted ['dit', 'via', 'pon', 'pip', 'his', 'pip', 'apa', 'ped', 'xas', 'sex']
Actual ['dit', 'via', 'led', 'fip', 'nub', 'pop', 'ara', 'ped', 'mas', 'sex']
```

```
english_words[8]
```

```
'til'
```

```
keras_model(braille_images[8])
```

```
'jil'
```

```
model_function(braille_images[8],5)
```

```
'til'
```

RESULTS

Keras Model

```
accuracy_k = sum([keras_output[i] == english_words[i] for i in range(len(english_words))])/len(english_words)
indexes_k = [i for i in range(len(english_words)) if keras_output[i] == english_words[i]]
correct_k = [keras_output[i] for i in indexes_k]
accuracy_k
```

```
0.06651198762567673
```

Keras Model + Language Model

```
accuracy = sum([output[i] == english_words[i] for i in range(len(english_words))])/len(english_words)
indexes = [i for i in range(len(english_words)) if output[i] == english_words[i] ]
correct = [output[i] for i in indexes]
accuracy
```

```
0.17092034029389017
```

Thanks for listening!