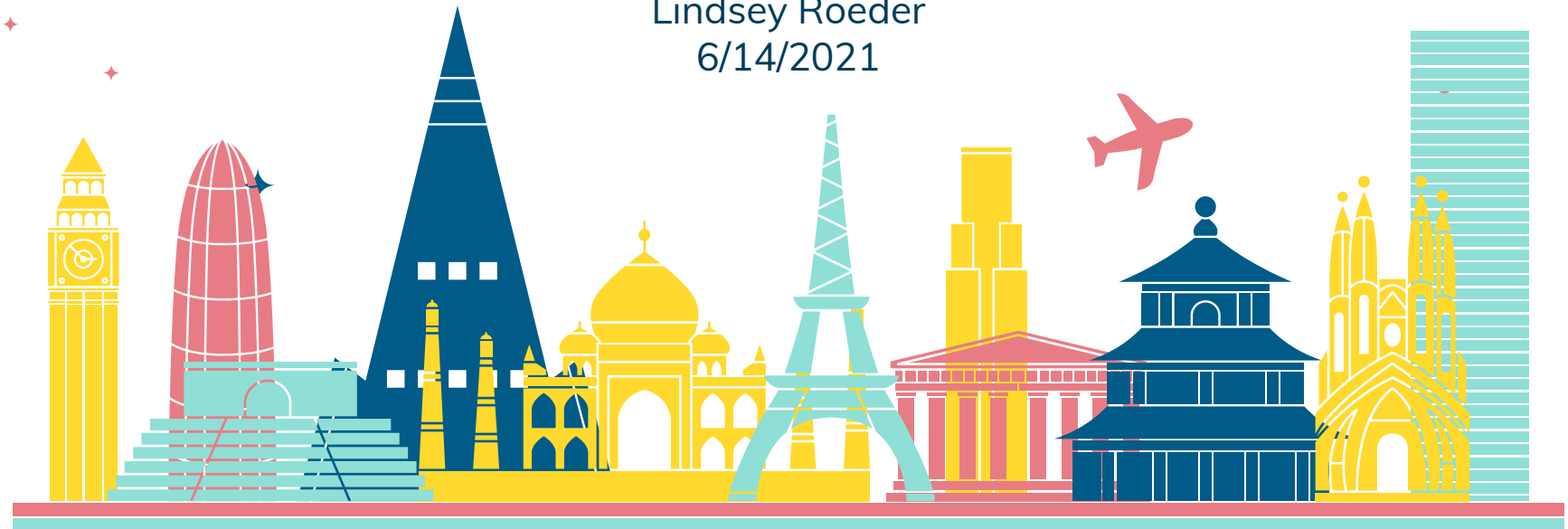
Building a Translator App with Image Classification

Lindsey Roeder
6/14/2021



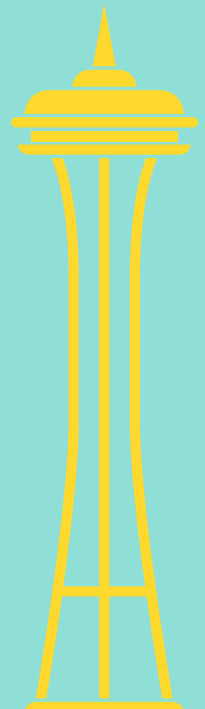
Problem Statement:

Can I build an image classification model that will generate a translation to any language from an uploaded picture?

“The art of translation lies less in knowing the language than in knowing your own”

— Mark Twain

Table of Contents



Collecting Data

from multiple sources

01

Data Sources

- Standard datasets
- Image search APIs
- Additional EDA & data cleaning

kaggle

Bing

Google

30 words

chosen from common items

30k+ images

between 800-1200 per word

3 GB

slow load and model time

02 Modeling

using neural networks

Modeling Layers

#1

Architecture

Xception

#2

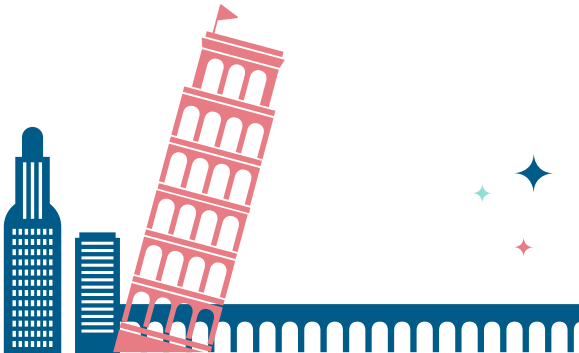
Pooling

GlobalAveragePooling2D

#3

Dense Layers

One Hidden



Evaluation

with accuracy

03

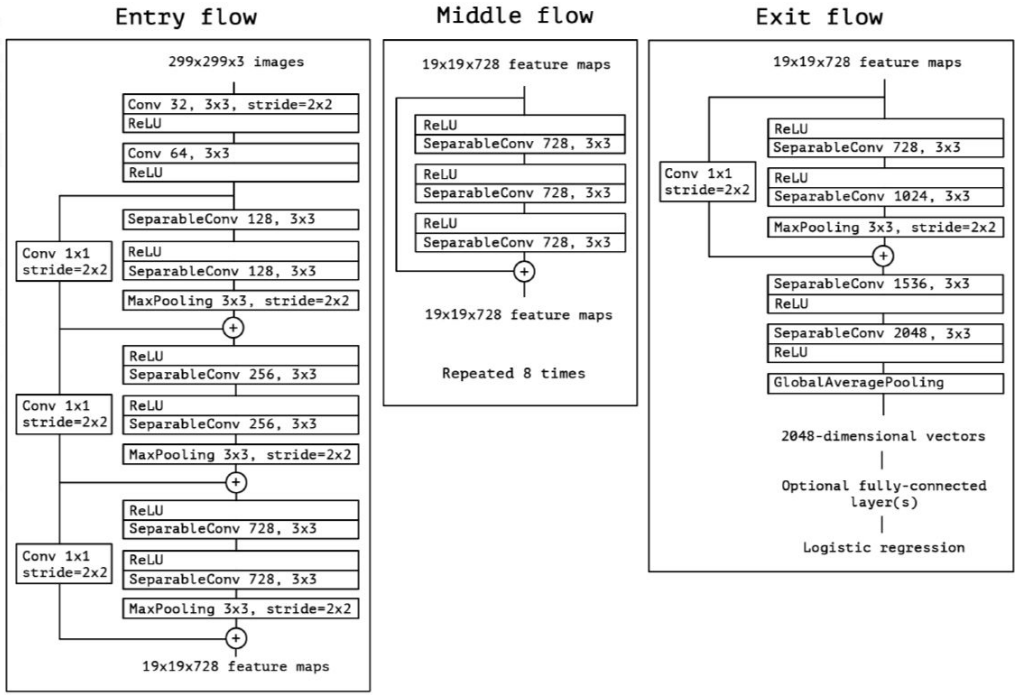
Scores by word Count

	Accuracy Score	Training Time
5 Words	96%	Low
15 Words	91%	Medium
30 Words	86%	High

Xception

- ImageNet dataset
- 36 convolutional layers
- Improved from Inception

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
global_average_pooling2d_2 ( (None, 2048)              0
_________________________________________________________________
dense_4 (Dense)              (None, 100)               204900
_________________________________________________________________
dense_5 (Dense)              (None, 30)                3030
=================================================================
Total params: 207,930
Trainable params: 207,930
Non-trainable params: 0
_________________________________________________________________
```



https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568

04 Translating

Google translator API

Google Translate

Text | Documents

DETECT LANGUAGE | ENGLISH | SPANISH | FRENCH

FRENCH | ENGLISH | SPANISH

The app translates to 107 languages!

36 / 5000

L'application se traduit en 107 langues !

Next Steps & Further Research

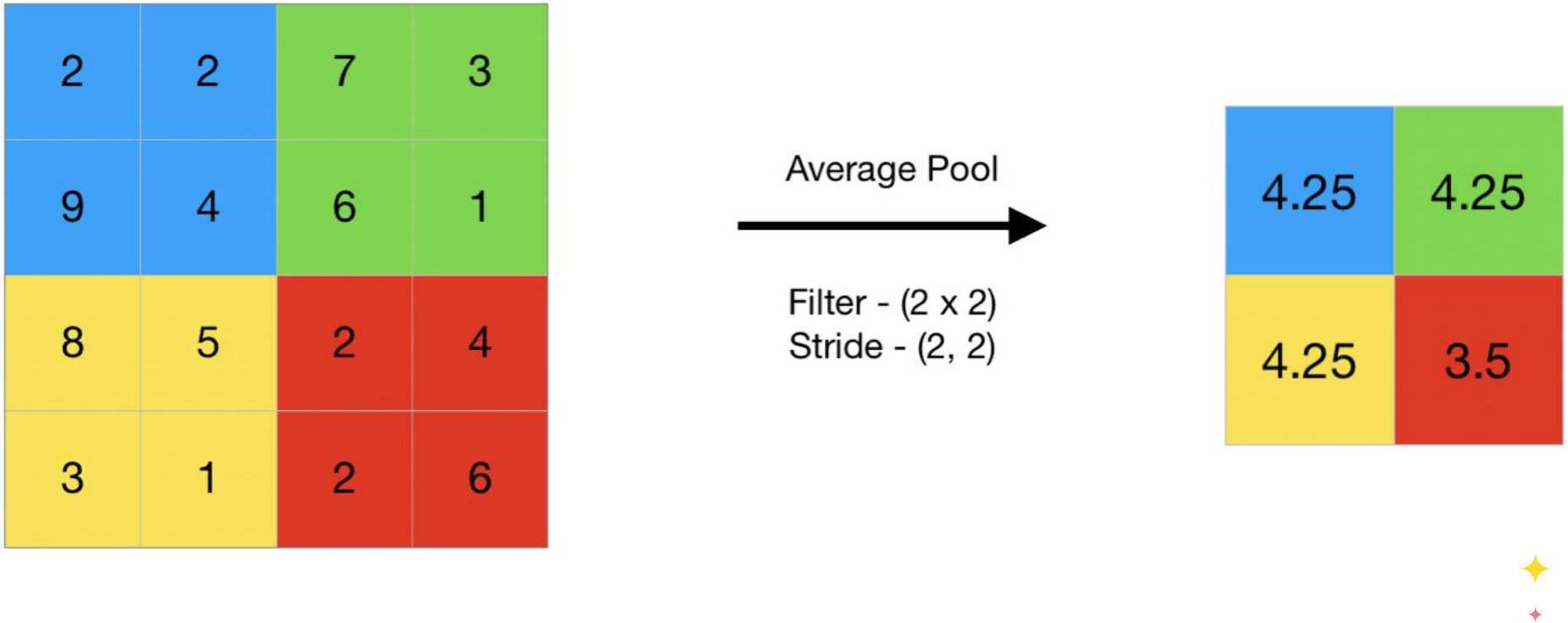
- Add more words
- Go through images
- Object detection
- Choose origin language

Thanks!

Questions?

Appendix

Global Average Pooling



Neural Net with Xception, convolution base used to predict triaining data before adding to model fit.

```
[ ] from tensorflow.keras.applications import MobileNetV2, VGG16, InceptionV3, Xception
```

```
[ ] pre_trained_model = Xception(
        include_top=False,
        weights='imagenet',
        input_shape=(150, 150, 3))

     # the top is the part that we don't need, the classification - we will set our own
```

```
[ ] train_preds = pre_trained_model.predict(train_data)
    val_preds = pre_trained_model.predict(val_data)
```

```
[ ] y_train = train_data.labels
    y_val = val_data.labels
```

```
[ ] y_train_categories = tf.keras.utils.to_categorical(y_train)
    y_val_categories = tf.keras.utils.to_categorical(y_val)
```

```
[ ] #since we didn't include the given top, we make our own and stick it to the bottom
    model = Sequential()

    model.add(GlobalAveragePooling2D())
    model.add(Dense(100, activation = 'relu'))

    model.add(Dense(30, activation = 'softmax'))
```

```
[ ] model.compile(loss = 'categorical_crossentropy',
                  optimizer = 'adam',
                  metrics = ['accuracy'])
```

```
[ ] history = model.fit(train_preds, y_train_categories,
                        validation_data=(val_preds, y_val_categories),

                        epochs = 15)
```

30 Word Model Code & Summary

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
global_average_pooling2d_2 ( (None, 2048)              0
_________________________________________________________________
dense_4 (Dense)              (None, 100)               204900
_________________________________________________________________
dense_5 (Dense)              (None, 30)                3030
=================================================================
Total params: 207,930
Trainable params: 207,930
Non-trainable params: 0
_________________________________________________________________
```

```
# results for validation set with 30 word classes
model.evaluate(val_preds, y_val_categories)
```

```
190/190 [==============================] - 1s 5ms/step - loss: 0.8229 - accuracy: 0.8550
[0.82290118932724, 0.8549555540084839]
```