Using CNNs for Painting Theme Comparison

Christina Hung and Nick Troccoli – *Stanford University*

Background/Introduction

- **We are investigating painting theme analysis**; Specifically, comparing the themes of two paintings.
- **Application:** clustering paintings based on thematic similarity, e.g. image categorization (social media, curation)
- **Application:** better understanding signature qualities of painting themes.

Project Statement

- **Goal**: Given an input pair, detect whether the pair has the same (0) or different (1) theme
- **Approach**: VGGNet-16 as a baseline architecture, with additional input featurizers
  - Color Histogram (RGB channels)
  - Painting Title Word Vector Featurizer
  - Object Detection Featurizer (IBM BlueMix)
- **Evaluation:** Compare validation and test set accuracy over multiple epochs

Dataset

- **WikiArt Dataset** (~35,750 images provided)
  - thematic labels by ChezMana
  - **10 themes**; ~400 samples per theme
  - to achieve **50-50 SAME-DIFFERENT thematic pairings**, uniformly sampled 320K paintings, half SAME and half DIFFERENT
  - divided into train, eval, and test sets

1. 

2. 

Model

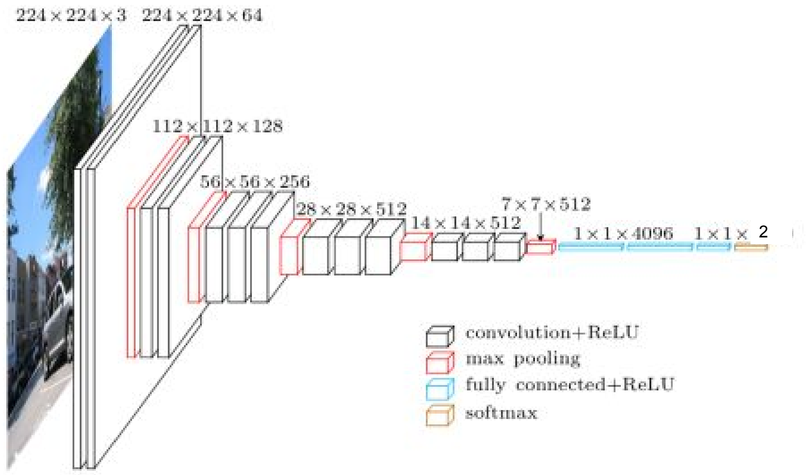


Figure 1: Macroarchitecture of VGG16 [1]

- **Painting Pairs Input:**
  - Stack the painting pairs over the color channel dimension, yielding a 224 x 224 x 6 image
  - Add initial conv layer (3 7x7 filters) to produce 224 x 224 x 3 input for VGGNet

All features are inputted via an additional size-224 dense layer, the output of which is added to the above input.

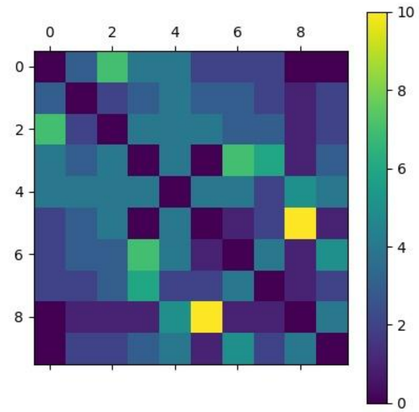
- **Color Histogram**
  - Create 3x16 per-channel histogram

- **Painting Titles Word Vector Featurizer**
  - Convert title to truncated/padded 5-word vector
  - I.e. for #2: ['evening', 'on', 'holy', 'saturday', '<END>']

- **Object Detection Featurizer**
  - Fetch a word vector (max 10 words) of objects using the IBM BlueMix Visual Recognition API
  - i.e. for #2: ['wagon', 'horse', 'men', 'rider' ..., 'crowd']

Results

- **BaselineModel**: ~52% test accuracy on 32,000



Key:
0 = 'roads-and-vehicles',
1 = 'forests-and-trees',
2 = 'houses-and-buildings',
3 = 'male-portraits',
4 = 'mountains',
5 = 'flowers-and-plants',
6 = 'fruits-and-vegetables',
7 = 'female-portraits',
8 = 'animals',
9 = 'boats-and-ships'

- **HistogramModel** performed about the same as above
- **TitlesModel**

--	lr=0.01	lr=0.1	lr=0.5
train_acc	0.556	0.510	0.510
val_acc	0.489	0.514	0.512

- **ObjectDetectionModel** performed about the same as TitleModel, ~51.4% accuracy on the val/test sets

Conclusion

- We attribute the similar performance across models to the incorporation of features only at the first layer, as well as the difficulty in applying convNets to more abstract paintings.
- **Future Work:**
  - extend featurized model to use a fusion of features
  - interleave featurizers throughout layers of network

Sources

[1] Frossard, D. VGG in Tensorflow, https://www.cs.toronto.edu/~frossard/post/vgg16/
[2] IBM Bluemix, https://console.ng.bluemix.net/