Investigating a New Approach to Classifying Geospatial Polygons

DATS 6203 – Machine Learning II, Group 1
Dan Vitale, Limin Tan, Nathan Justice

Roadmap

1 Introduction

1. Background
2. Project Goal

2 Literature Review

1. Related Work
2. Existing Framework
3. Our Framework
4. Comparisons

3 Data

1. Benchmarks
2. Dataset Descriptions
3. Dataset Preprocessing

4 Modeling

1. 2D CNN
2. Pre-Trained Models

5 Conclusion

1. Take-Aways
2. Future Work

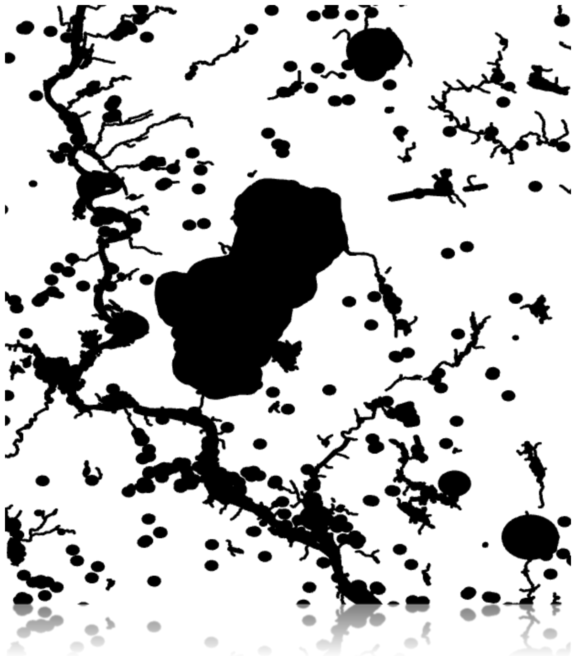
- Countless applications of GIS
- Common experience dealing with unsourceable data

The goal of our project is to develop a deep learning framework for classifying spatial polygons, based solely on their geometry, that is more flexible, light-weight, and accurate than existing frameworks.

Literature Review

- Growing popularity of machine learning in GIS
- Advantages of deep learning

Polygon



Raster

Cornell University

arXiv.org > stat > arXiv:1806.03857

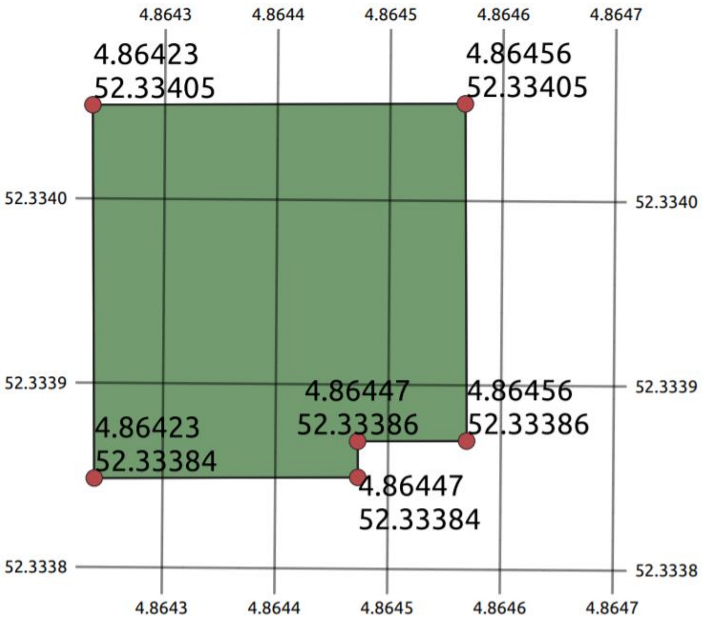
Statistics > Machine Learning

Deep Learning for Classification Tasks on Geospatial Vector Polygons

Rein van 't Veer, Peter Bloem, Erwin Folmer

(Submitted on 11 Jun 2018 (v1), last revised 11 Jun 2019 (this version, v2))

"Can deep learning models achieve accuracies comparable with shallow learning models in analysing geospatial vector shapes?"



Polygon coordinates	Center: remove mean of [4.8644271, 52.3339057]	Scale: divide by scale factor of 2.64501e-4
4.86447, 52.33384	4.2857e-5, -6.5714e-5	0.16198, -0.24845
4.86447, 52.33386	4.2857e-5, -4.5714e-5	0.16198, -0.17283
4.86456, 52.33386	1.32857e-4, -4.5714e-5	0.50229, -0.17283
4.86456, 52.33386	1.32857e-4, 1.44286e-4	0.50229, 0.54550
4.86423, 52.33405	-1.97143e-4, 1.44286e-4	-0.74534, 0.54550
4.86423, 52.33405	-1.97143e-4, -6.5714e-5	-0.74534, -0.24845
4.86447, 52.33384	4.2857e-5, -6.5714e-5	0.16959, -0.24845

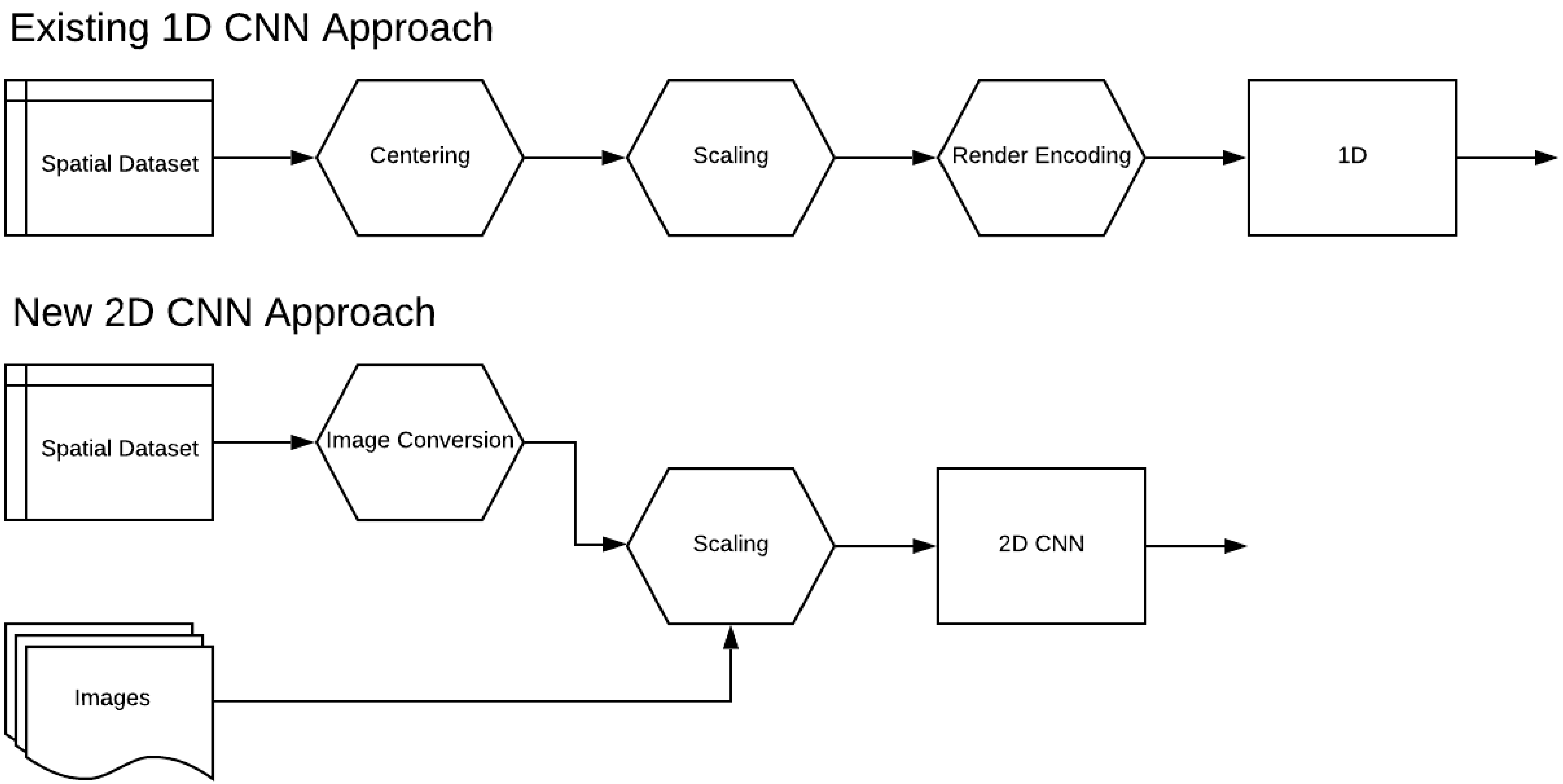
(b)

Tensor representation
[0.16198, -0.24845, 1, 0, 0],
[0.16198, -0.17283, 1, 0, 0],
[0.50229, -0.17283, 1, 0, 0],
[0.50229, 0.54550, 1, 0, 0],
[-0.74534, 0.54550, 1, 0, 0],
[-0.74534, -0.24845, 1, 0, 0],
[0.16959, -0.24845, 0, 0, 1]

Literature Review

1. Related Work
2. **Existing Framework**
3. Our Framework
4. Comparisons

- 2D CNN on images



Literature Review

Method	Task (no. of classes)	
	Neighbourhood inhabitants (2)	Building types (9)
Majority class	0.514	0.142
k-NN	0.671	0.377
Logistic regression	0.659	0.328
SVM RBF	**0.683**	0.365
Decision tree	0.682	0.389
CNN	0.664 ± 0.005	$\mathbf{0.408 \pm 0.003}$

Data

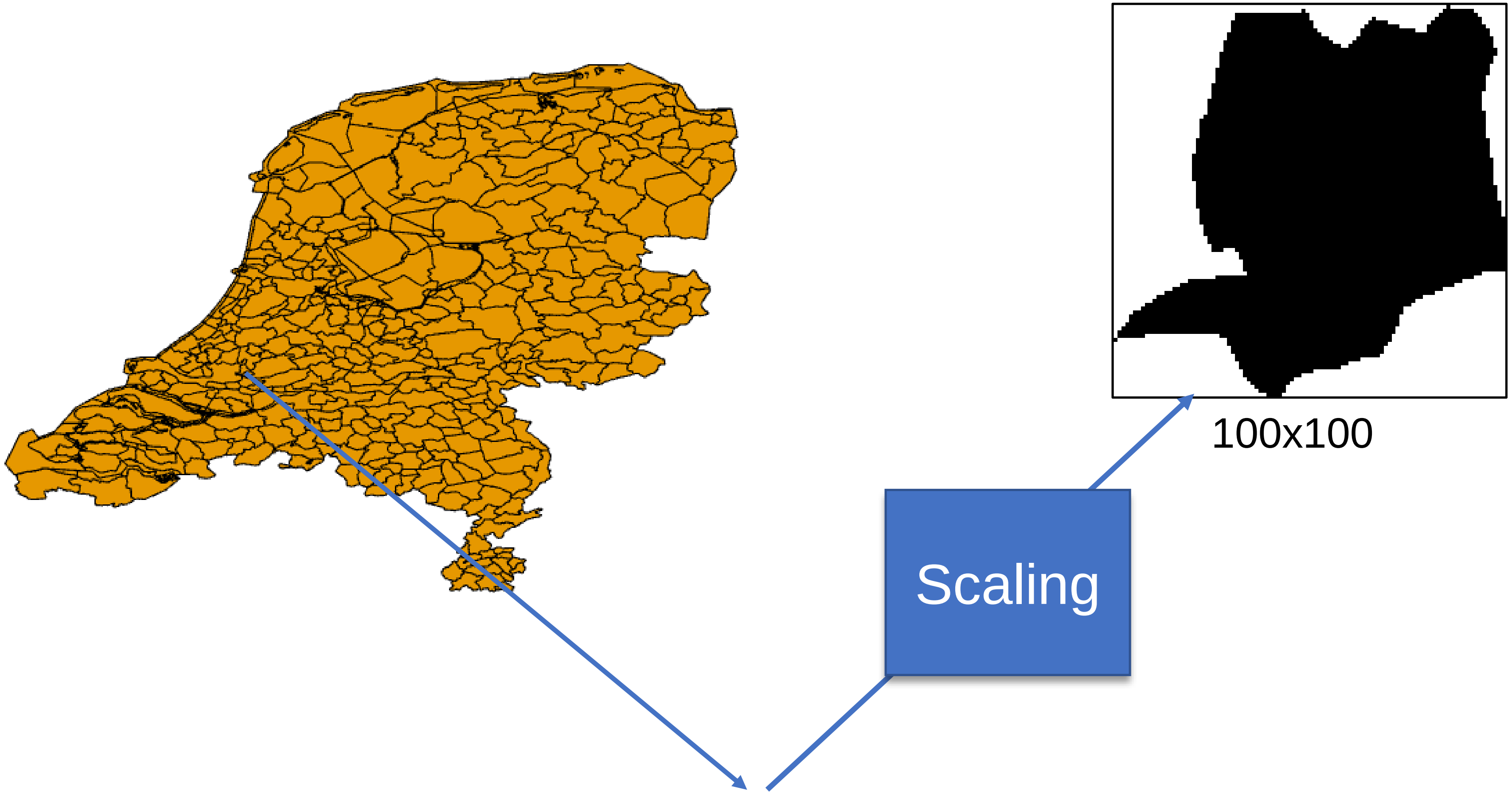
Neighbourhood inhabitants	
Class	frequency
$\geq$ median	6,610
$<$ median	6,598
Total	13,208

Buildings	
Function	frequency
Habitation	23,000
Industrial	23,000
Lodging	23,000
Shopping	23,000
Gatherings	22,007
Office	21,014
Education	10,717
Healthcare	7,832
Sports	6,916
Total	160,486

3

Data

1. Benchmarks
2. Dataset Descriptions
3. **Dataset Preprocessing**

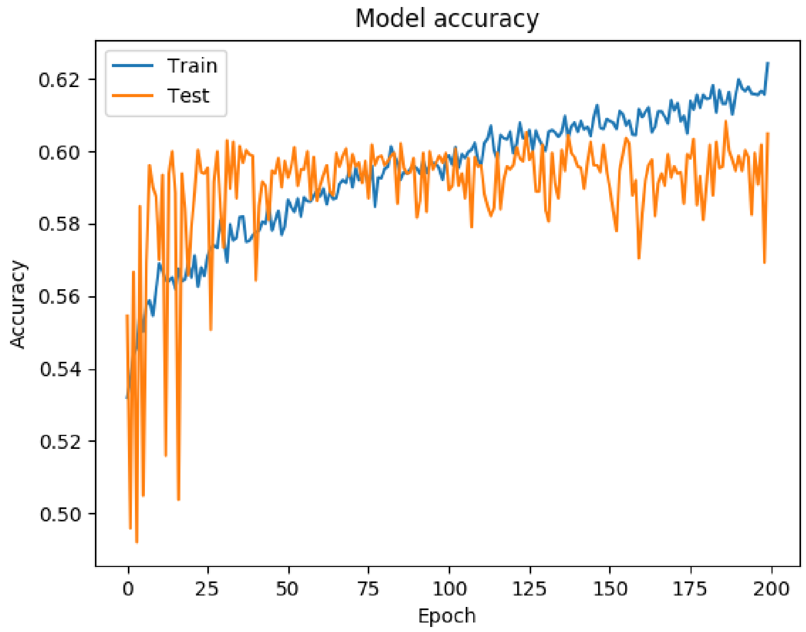


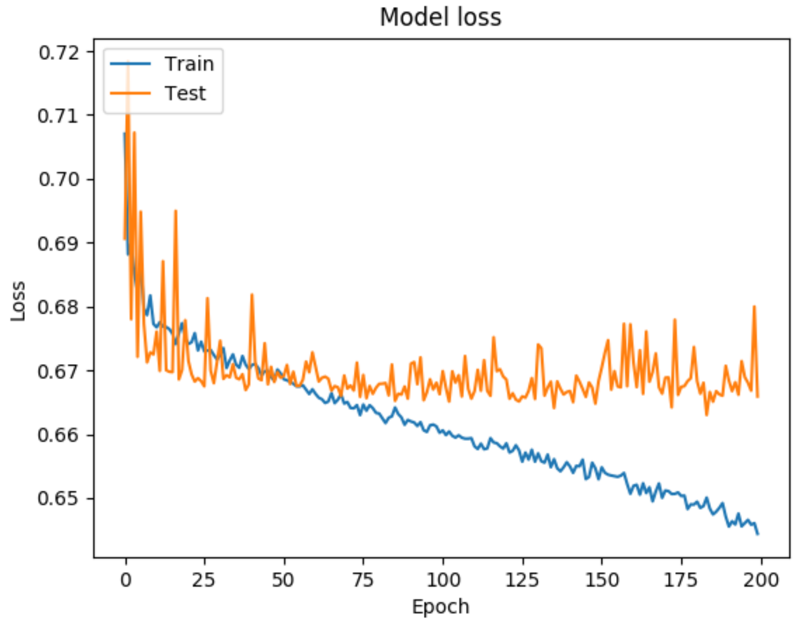
MULTIPOLYGON (((6.490848046163124 52.77088348471561, 6.490982951822918 52.7708789922233, 6.490998928293062 52.77087846443353, 6.491122124970168 52.77088414443787, 6.491215279514428 52.770888439578925, 6.491272190126054 52.77089106676583, 6.4920275320370235 52.77092589893195, 6.4923542897349655 52.770946675142994, 6.492480266246836 52.7709561924976, 6.49253288950031....

4

Network Architecture: Neighborhoods- Baseline Paper Comparison

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 100, 100, 32)      832
_________________________________________________________________
activation_1 (Activation)    (None, 100, 100, 32)      0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 34, 34, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 34, 34, 65)        52065
_________________________________________________________________
activation_2 (Activation)    (None, 34, 34, 65)        0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 65)                0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                2112
_________________________________________________________________
activation_3 (Activation)    (None, 32)                0
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 66
_________________________________________________________________
activation_4 (Activation)    (None, 2)                 0
=================================================================
Total params: 55,075
Trainable params: 55,075
Non-trainable params: 0
_________________________________________________________________
```





Modeling

1. **2D CNN**
2. Pre-Trained Models

Network Architecture: Neighborhoods- Baseline Paper Comparison

Classification Report

	precision	recall	f1-score	support
0	0.54	0.61	0.57	487
1	0.62	0.55	0.59	570
accuracy			0.58	1057
macro avg	0.58	0.58	0.58	1057
weighted avg	0.58	0.58	0.58	1057

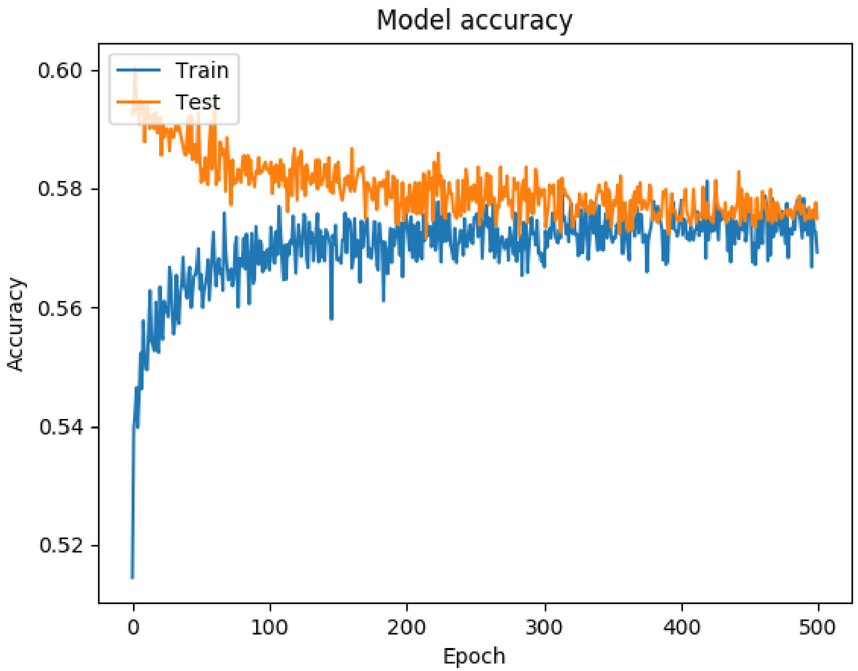
Acc: 0.5795

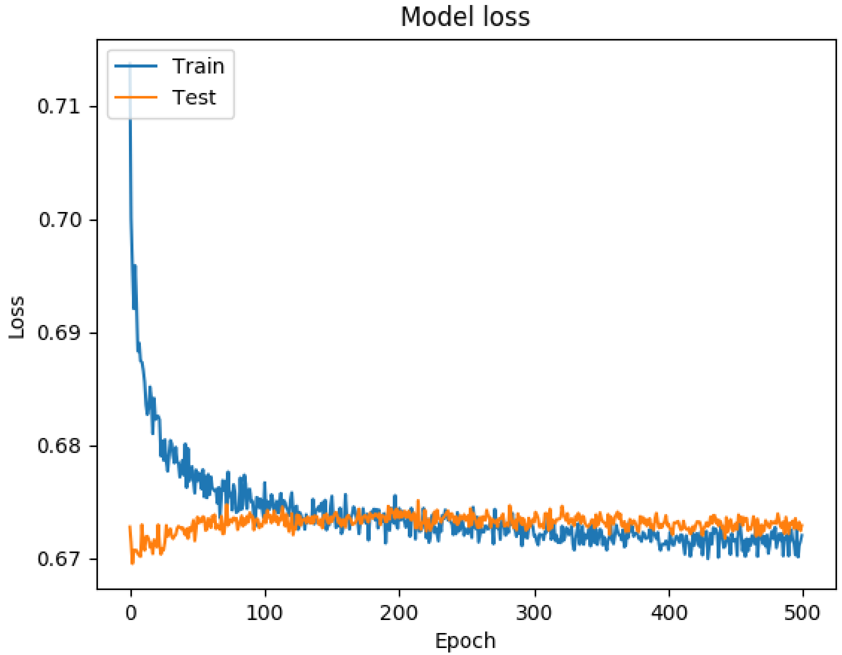
Loss: 0.6744

4

Network Architecture: Neighborhoods

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 96, 96, 32)        832
_________________________________________________________________
batch_normalization_1 (Batch (None, 96, 96, 32)        128
_________________________________________________________________
activation_1 (Activation)    (None, 96, 96, 32)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 32)        0
_________________________________________________________________
spatial_dropout2d_1 (Spatial (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 28, 28, 64)        51264
_________________________________________________________________
batch_normalization_2 (Batch (None, 28, 28, 64)        256
_________________________________________________________________
activation_2 (Activation)    (None, 28, 28, 64)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 10, 10, 64)        0
_________________________________________________________________
spatial_dropout2d_2 (Spatial (None, 10, 10, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 6, 6, 128)         204928
_________________________________________________________________
batch_normalization_3 (Batch (None, 6, 6, 128)         512
_________________________________________________________________
activation_3 (Activation)    (None, 6, 6, 128)         0
_________________________________________________________________
spatial_dropout2d_3 (Spatial (None, 6, 6, 128)         0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 700)               90300
_________________________________________________________________
activation_4 (Activation)    (None, 700)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 700)               0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 1402
_________________________________________________________________
activation_5 (Activation)    (None, 2)                 0
=================================================================
Total params: 349,622
Trainable params: 349,174
Non-trainable params: 448
_________________________________________________________________
```





Modeling

1. **2D CNN**
2. Pre-Trained Models

Network Architecture: Neighborhoods

Classification Report

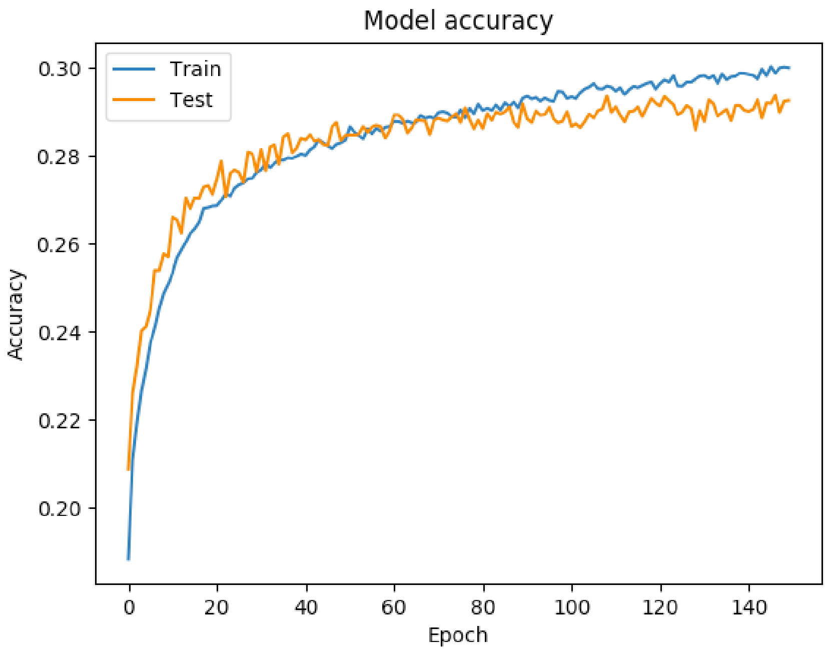
	precision	recall	f1-score	support
0	0.49	0.91	0.64	487
1	0.71	0.19	0.29	570
accuracy			0.52	1057
macro avg	0.60	0.55	0.47	1057
weighted avg	0.61	0.52	0.45	1057

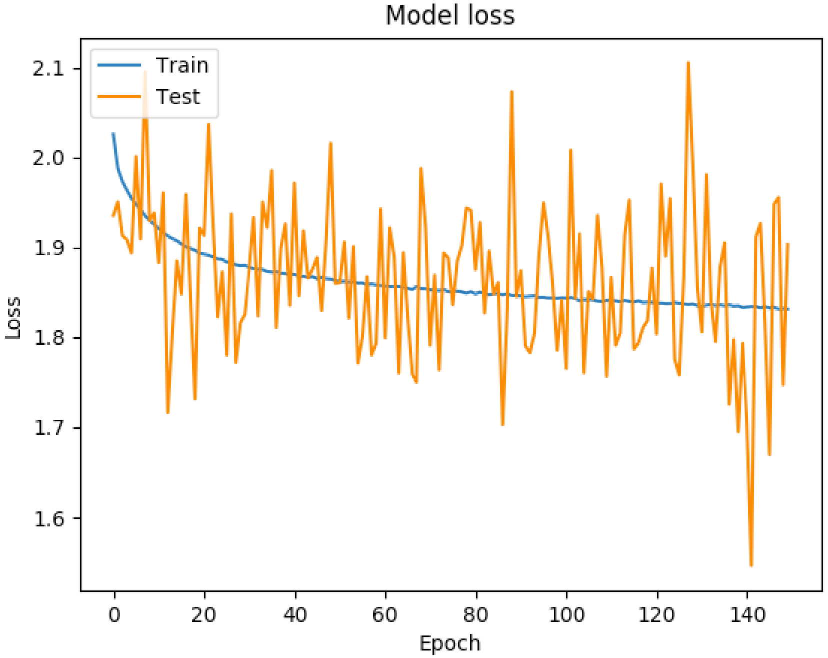
Acc: 0.5199

Loss: 0.6854

Network Architecture: Buildings Dataset

```
=================================================================
conv2d_1 (Conv2D)            (None, 96, 96, 16)        416
_________________________________________________________________
batch_normalization_1 (Batch (None, 96, 96, 16)        64
_________________________________________________________________
activation_1 (Activation)    (None, 96, 96, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 19, 19, 16)        0
_________________________________________________________________
spatial_dropout2d_1 (Spatial (None, 19, 19, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 15, 15, 32)        12832
_________________________________________________________________
batch_normalization_2 (Batch (None, 15, 15, 32)        128
_________________________________________________________________
activation_2 (Activation)    (None, 15, 15, 32)        0
_________________________________________________________________
average_pooling2d_1 (Average (None, 3, 3, 32)          0
_________________________________________________________________
spatial_dropout2d_2 (Spatial (None, 3, 3, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 288)               0
_________________________________________________________________
dense_1 (Dense)              (None, 700)               202300
_________________________________________________________________
activation_3 (Activation)    (None, 700)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 700)               0
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 5608
_________________________________________________________________
activation_4 (Activation)    (None, 8)                 0
=================================================================
Total params: 221,348
Trainable params: 221,252
Non-trainable params: 96
_________________________________________________________________
```





Modeling

1. **2D CNN**
2. Pre-Trained Models

Results: Buildings Dataset

Classification Report

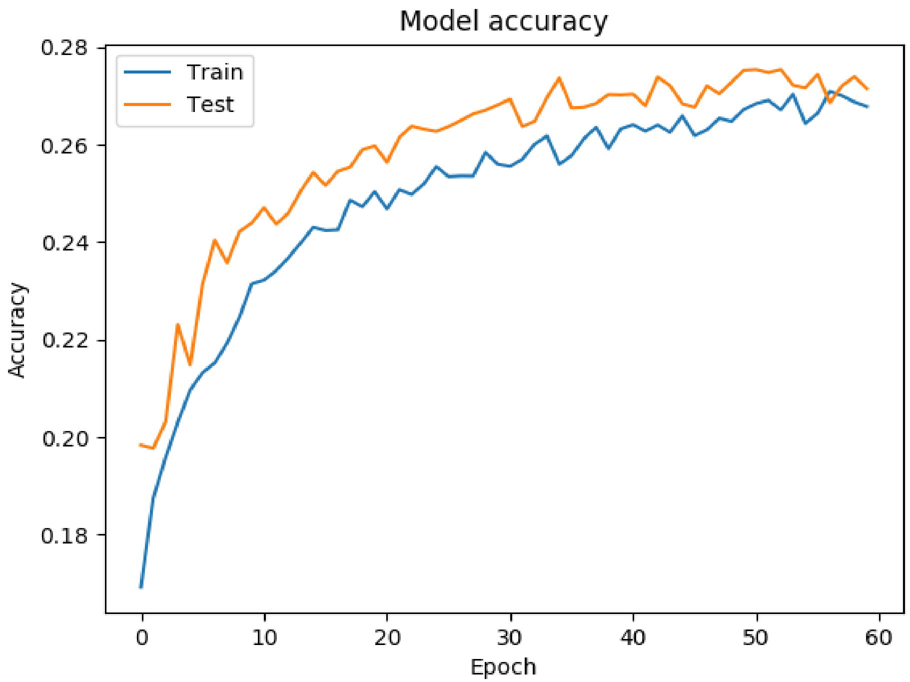
	precision	recall	f1-score	support
0	0.32	0.63	0.42	830
1	0.34	0.09	0.14	1789
2	0.34	0.49	0.40	1802
3	0.10	0.13	0.11	637
4	0.25	0.31	0.28	1845
5	0.30	0.47	0.36	1820
6	0.19	0.03	0.06	1698
7	0.31	0.22	0.26	1884
accuracy			0.29	12305
macro avg	0.27	0.30	0.25	12305
weighted avg	0.28	0.29	0.26	12305

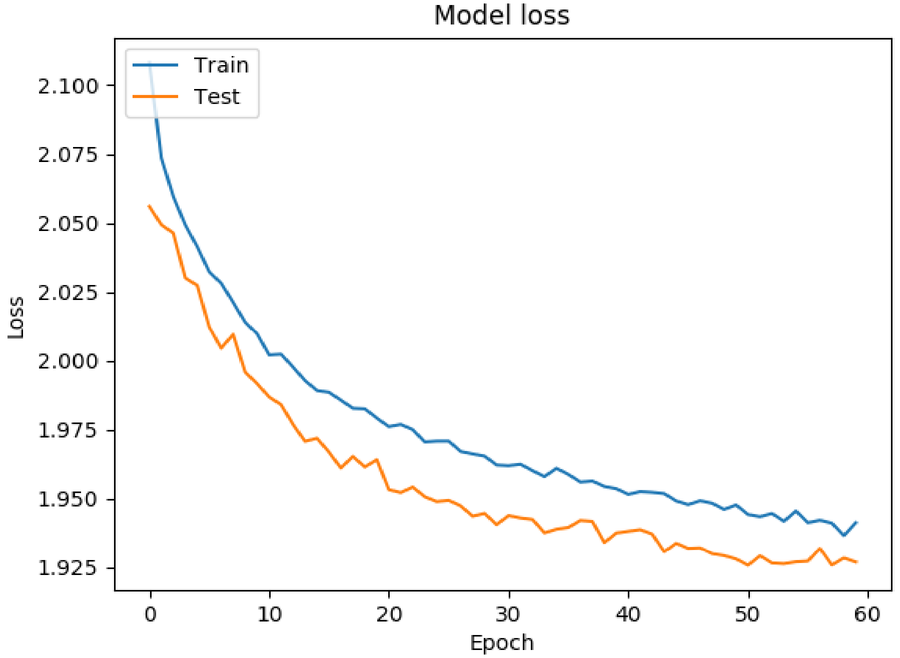
acc: 0.2879

loss: 1.8293

Network Architecture: Buildings Dataset

```
=================================================================================
conv2d_1 (Conv2D)            (None, 91, 91, 32)        3232
_________________________________________________________________________________
batch_normalization_1 (Batch (None, 91, 91, 32)        128
_________________________________________________________________________________
activation_1 (Activation)    (None, 91, 91, 32)        0
_________________________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 18, 18, 32)        0
_________________________________________________________________________________
spatial_dropout2d_1 (Spatial (None, 18, 18, 32)        0
_________________________________________________________________________________
conv2d_2 (Conv2D)            (None, 14, 14, 64)        51264
_________________________________________________________________________________
batch_normalization_2 (Batch (None, 14, 14, 64)        256
_________________________________________________________________________________
activation_2 (Activation)    (None, 14, 14, 64)        0
_________________________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________________________
spatial_dropout2d_2 (Spatial (None, 7, 7, 64)          0
_________________________________________________________________________________
conv2d_3 (Conv2D)            (None, 5, 5, 128)         73856
_________________________________________________________________________________
batch_normalization_3 (Batch (None, 5, 5, 128)         512
_________________________________________________________________________________
activation_3 (Activation)    (None, 5, 5, 128)         0
_________________________________________________________________________________
average_pooling2d_1 (Average (None, 1, 1, 128)         0
_________________________________________________________________________________
spatial_dropout2d_3 (Spatial (None, 1, 1, 128)         0
_________________________________________________________________________________
flatten_1 (Flatten)          (None, 128)               0
_________________________________________________________________________________
dense_1 (Dense)              (None, 700)               90300
_________________________________________________________________________________
activation_4 (Activation)    (None, 700)               0
_________________________________________________________________________________
dropout_1 (Dropout)          (None, 700)               0
_________________________________________________________________________________
dense_2 (Dense)              (None, 9)                 6309
_________________________________________________________________________________
activation_5 (Activation)    (None, 9)                 0
=================================================================================
Total params: 225,857
Trainable params: 225,409
Non-trainable params: 448
_________________________________________________________________________________
```





4

Modeling

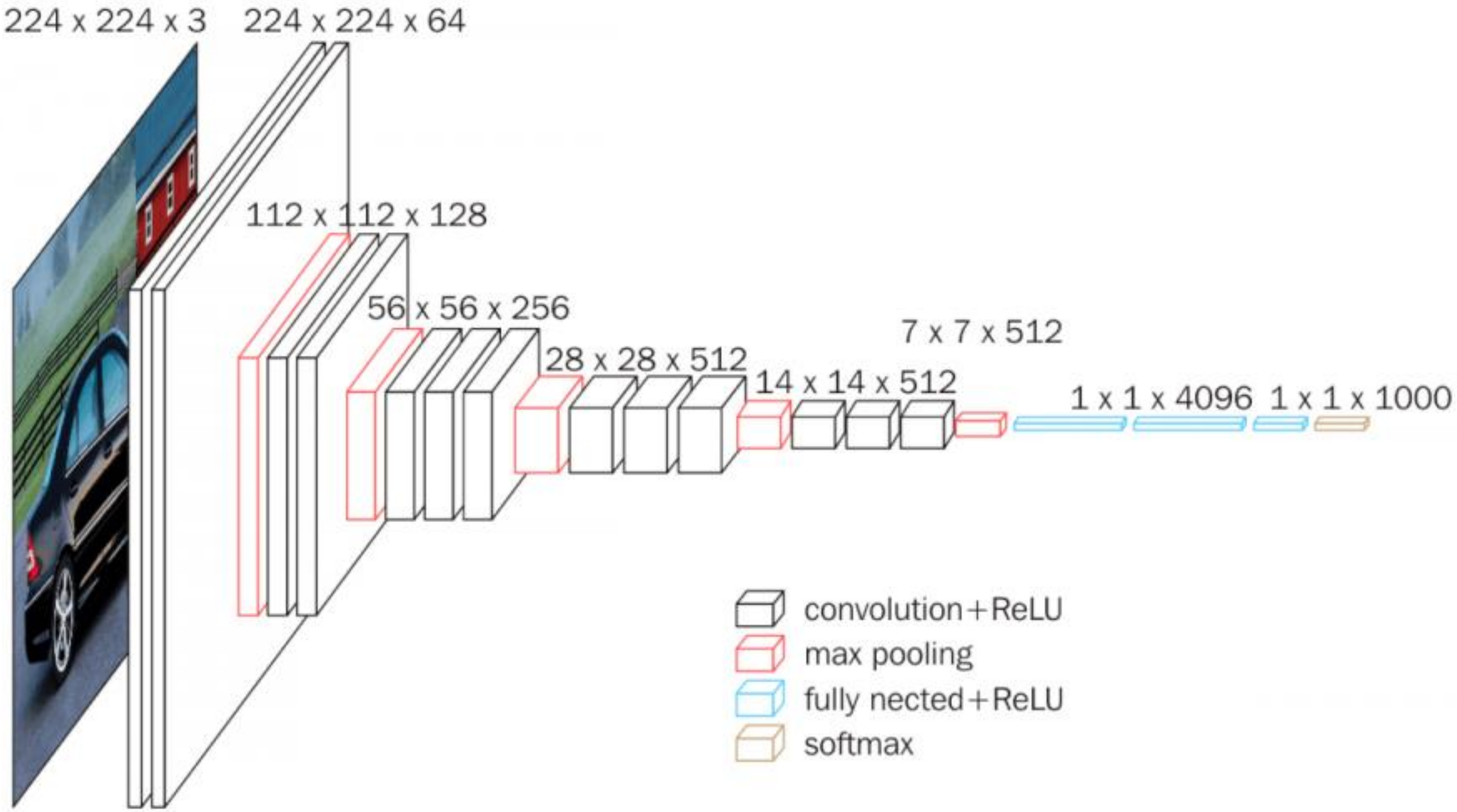
1. **2D CNN**
2. Pre-Trained Models

Network Architecture: Buildings Dataset

	precision	recall	f1-score	support
0	0.38	0.52	0.44	830
1	0.23	0.19	0.21	1789
2	0.34	0.42	0.37	1802
3	0.73	0.01	0.02	637
4	0.23	0.38	0.29	1845
5	0.27	0.41	0.32	1820
6	0.21	0.06	0.09	1698
7	0.26	0.22	0.24	1884
8	0.00	0.00	0.00	531
accuracy			0.27	12836
macro avg	0.29	0.25	0.22	12836
weighted avg	0.28	0.27	0.25	12836

loss: 3.1383

acc: 0.2734



VGG is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition” . The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes.

1.Prepare data

2.Extract features from the convolutional base

3. Fully-connected layers

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 100, 100, 3)	0
block1_conv1 (Conv2D)	(None, 100, 100, 64)	1792
block1_conv2 (Conv2D)	(None, 100, 100, 64)	36928
block1_pool (MaxPooling2D)	(None, 50, 50, 64)	0
block2_conv1 (Conv2D)	(None, 50, 50, 128)	73856
block2_conv2 (Conv2D)	(None, 50, 50, 128)	147584
block2_pool (MaxPooling2D)	(None, 25, 25, 128)	0
block3_conv1 (Conv2D)	(None, 25, 25, 256)	295168
block3_conv2 (Conv2D)	(None, 25, 25, 256)	590080
block3_conv3 (Conv2D)	(None, 25, 25, 256)	590080
block3_pool (MaxPooling2D)	(None, 12, 12, 256)	0
block4_conv1 (Conv2D)	(None, 12, 12, 512)	1180160
block4_conv2 (Conv2D)	(None, 12, 12, 512)	2359808
block4_conv3 (Conv2D)	(None, 12, 12, 512)	2359808
block4_pool (MaxPooling2D)	(None, 6, 6, 512)	0
block5_conv1 (Conv2D)	(None, 6, 6, 512)	2359808
block5_conv2 (Conv2D)	(None, 6, 6, 512)	2359808
block5_conv3 (Conv2D)	(None, 6, 6, 512)	2359808
block5_pool (MaxPooling2D)	(None, 3, 3, 512)	0

Total params: 14,714,688

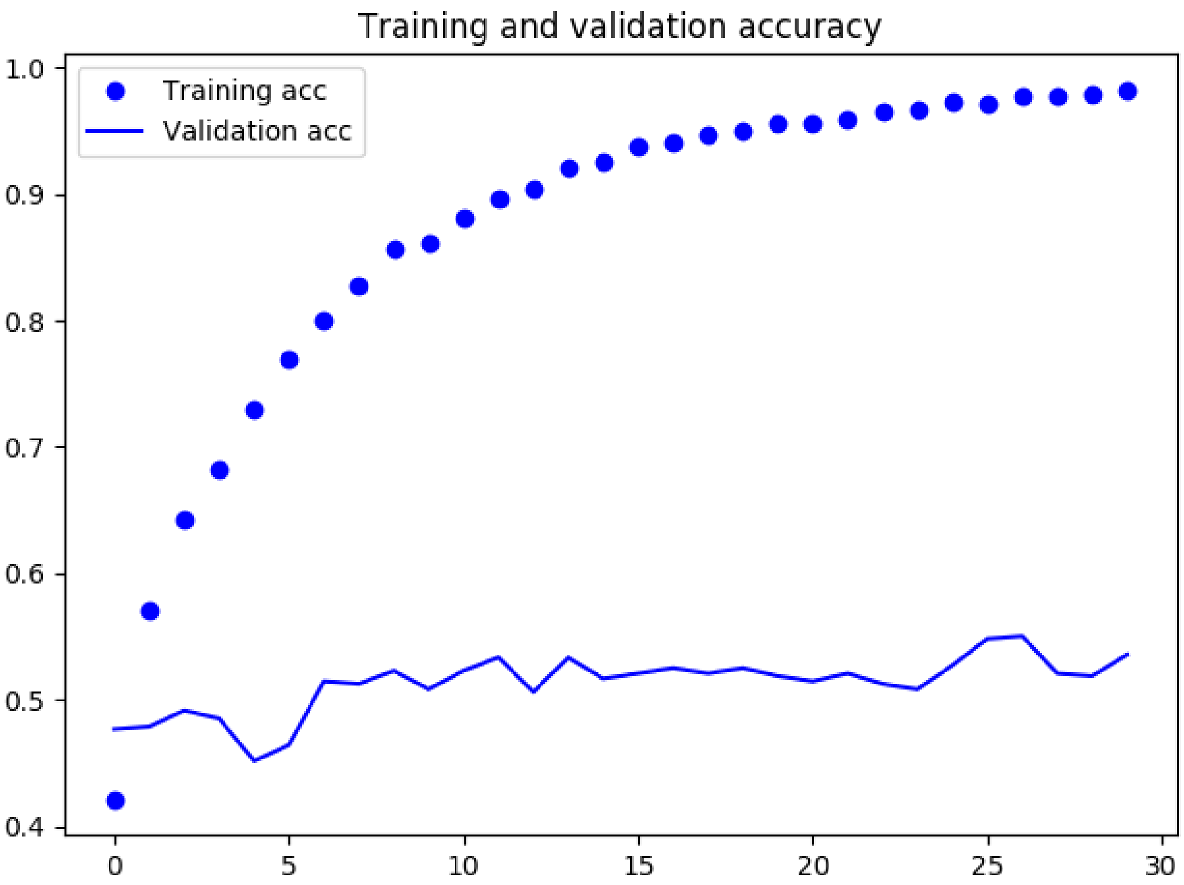
Modeling

1. 2D CNN
2. **Pre-Trained Models**

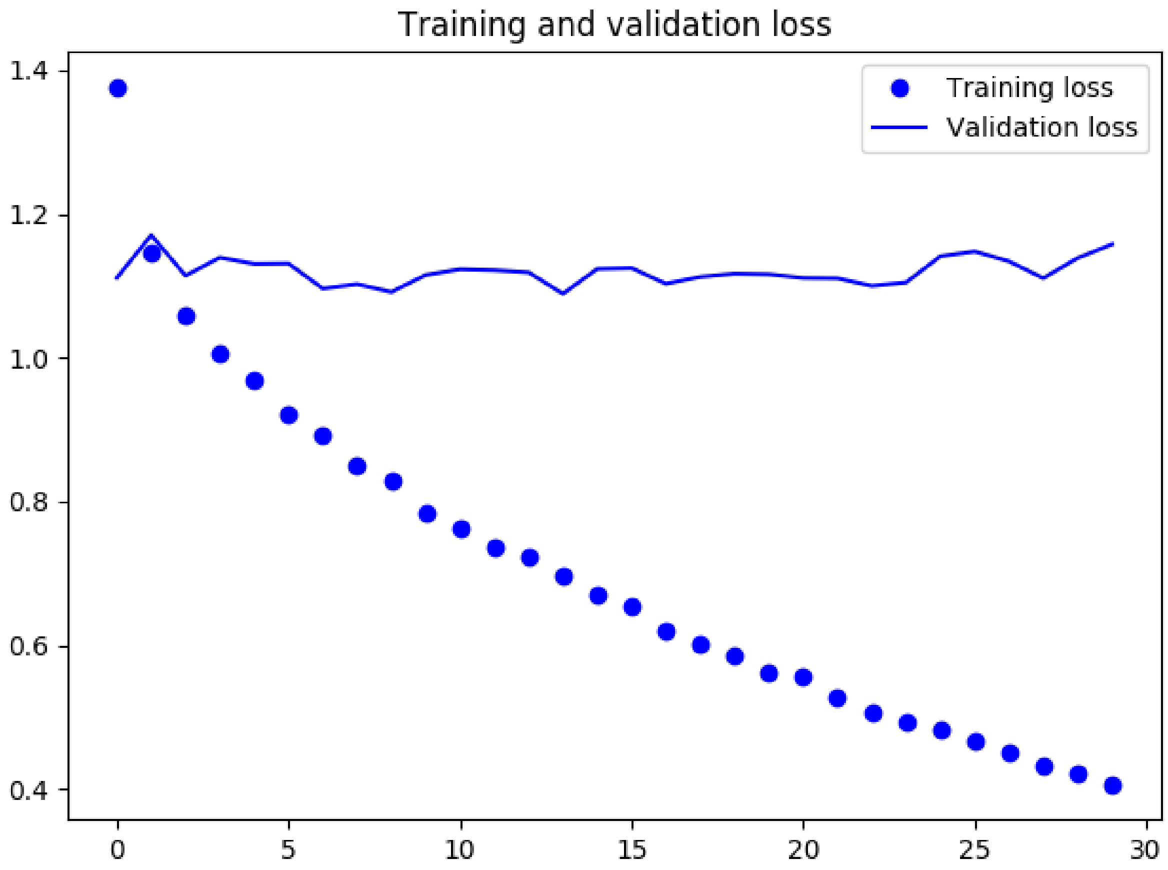
4

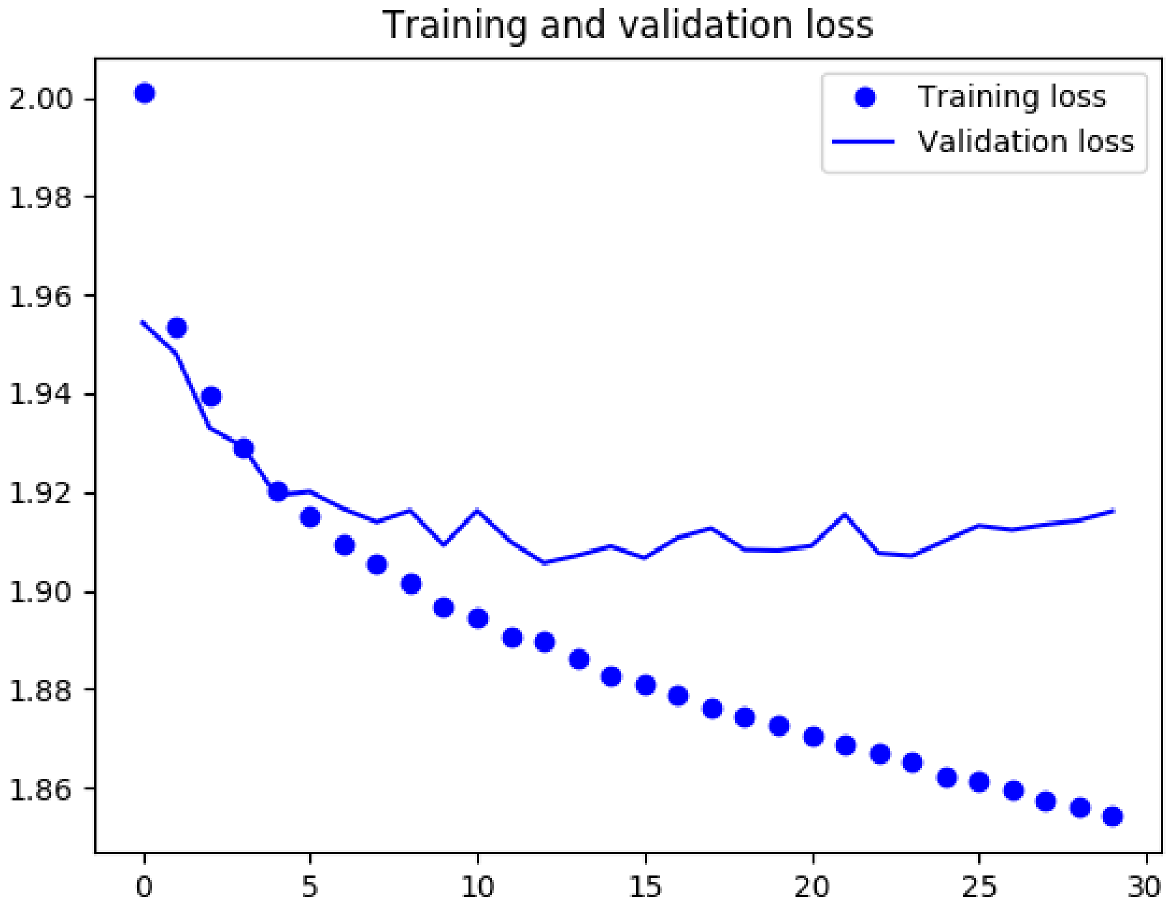
Modeling

1. 2D CNN
2. **Pre-Trained Models**



$$0.408 \pm 0.003$$

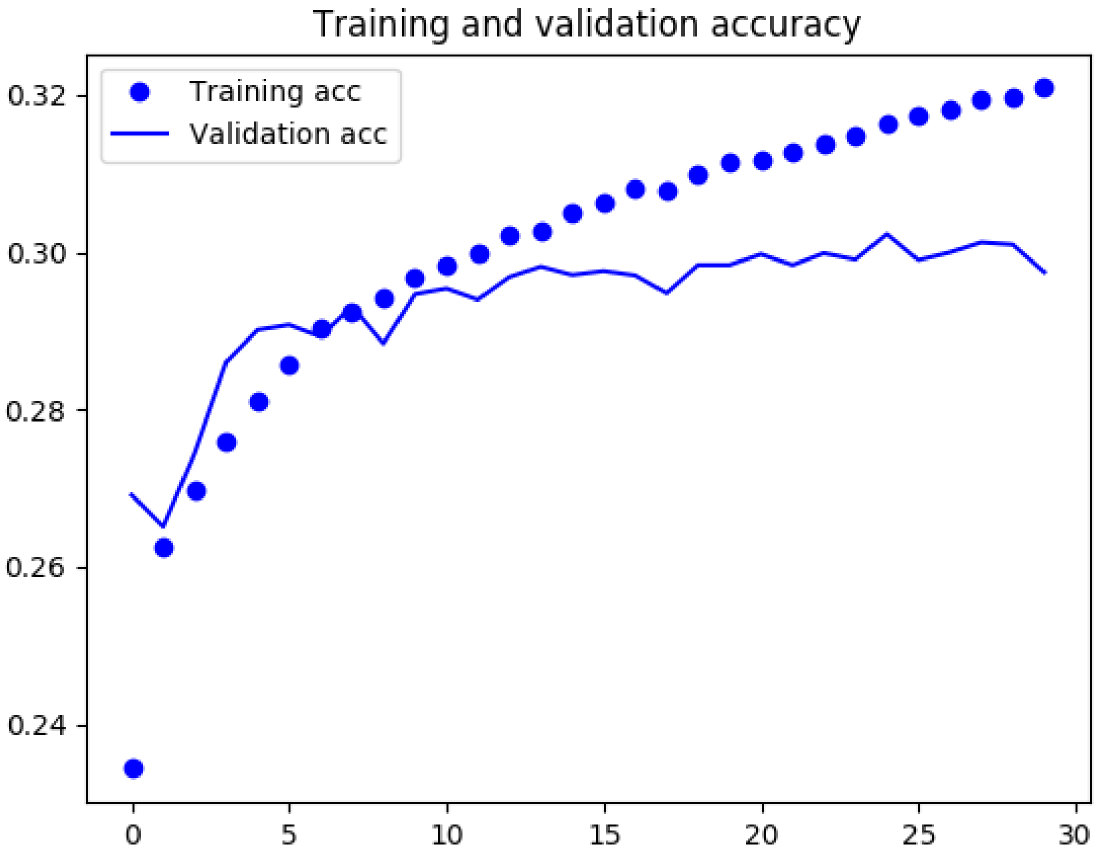




4

Modeling

1. 2D CNN
2. **Pre-Trained Models**



$$0.408 \pm 0.003$$

- Flexible, Light-Weight, Accurate
- New benchmarks for 2D CNN trained on geometry coordinates

- Combining other datasets to add more features for training


MY CODE DOESN'T WORK
I HAVE NO IDEA WHY
MY CODE WORKS
I HAVE NO IDEA WHY

?

Questions

Term	GIS Definition
Geometry	spatial representation of an object comprised of one or more points
Vector	geometry defined by vertices and edges
Feature	geospatial object
Shape	geospatial object geometry
Polygon	sequence of three or more connected lines
Multi-Polygon	feature instance with two or more polygons