Group 5: Pooling Strategies

Jacob Crabtree

Anthony Cavallero

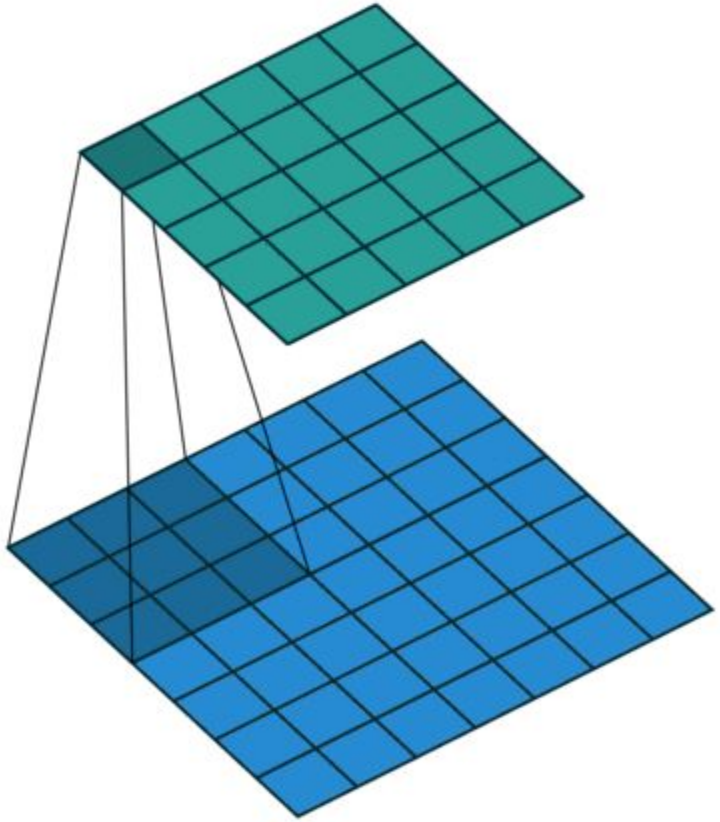
Jason Clemens

Roadmap:

1. Recap of Pooling/CNN
2. Hypotheses
3. Methodology
4. Results
5. Discussion
6. Conclusions and Future Work
7. Questions

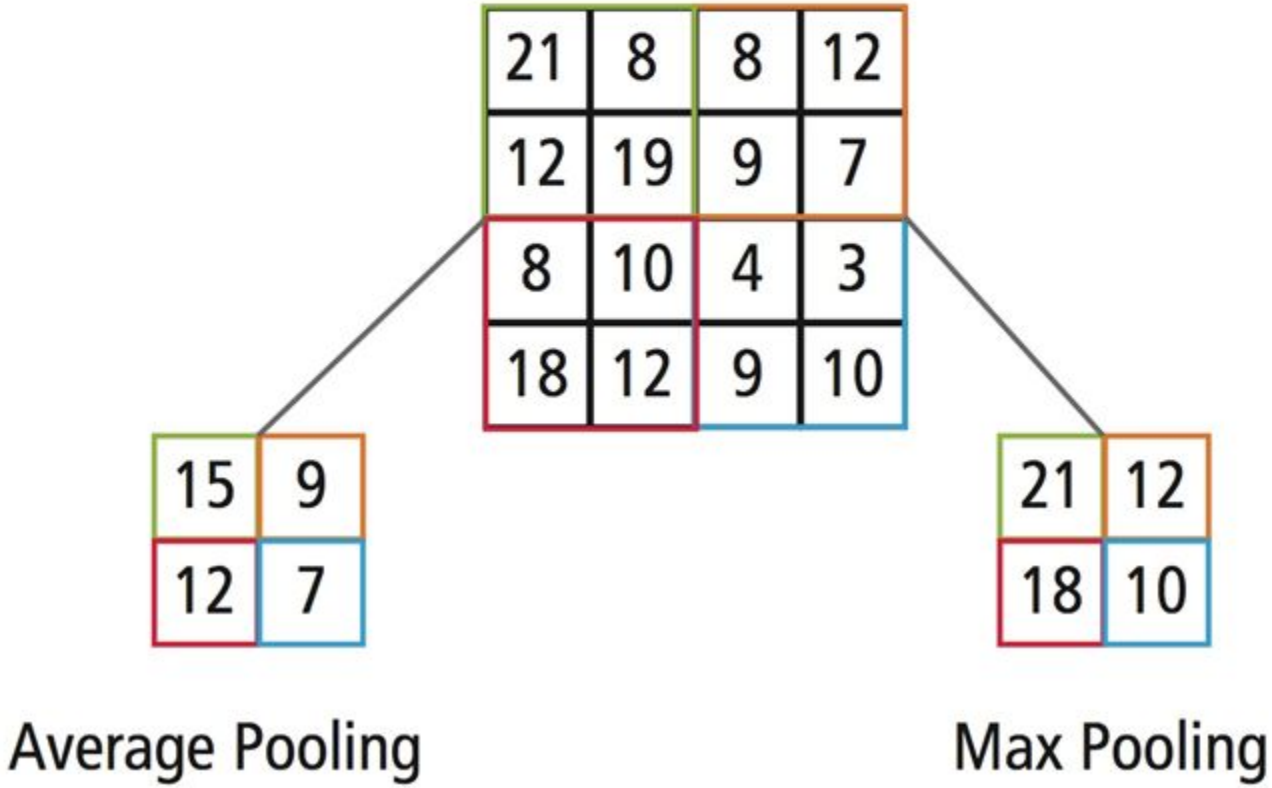
Refresher: What is Convolution?

- Move a kernel over input areas
- Linear Operation
- Used to extract image features usually
- Replaces dense layers



Refresher: What is Pooling?

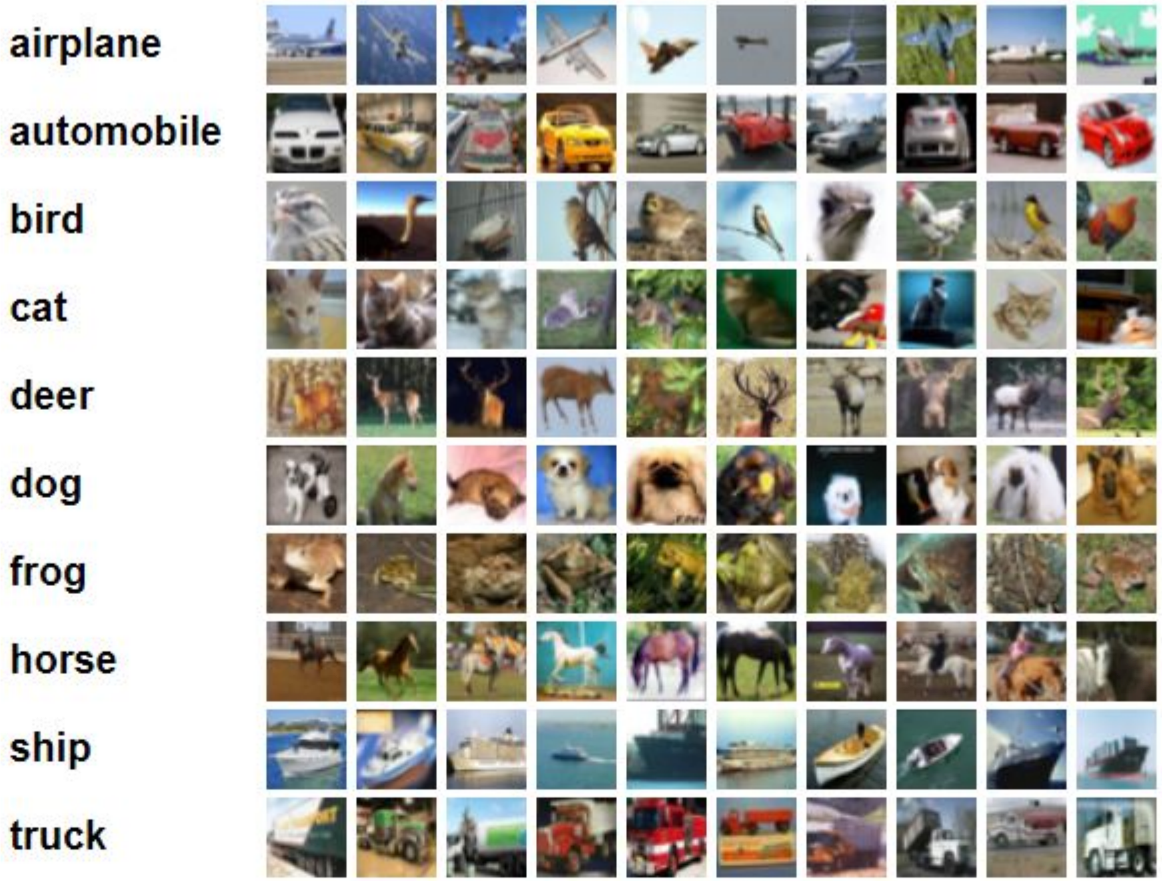
- Summarizes an area of input
- Non-linear operation
  - For Max pooling
- Loses information
  - Can be a desirable effect
- Very common in CNNs



[3]

Hypotheses

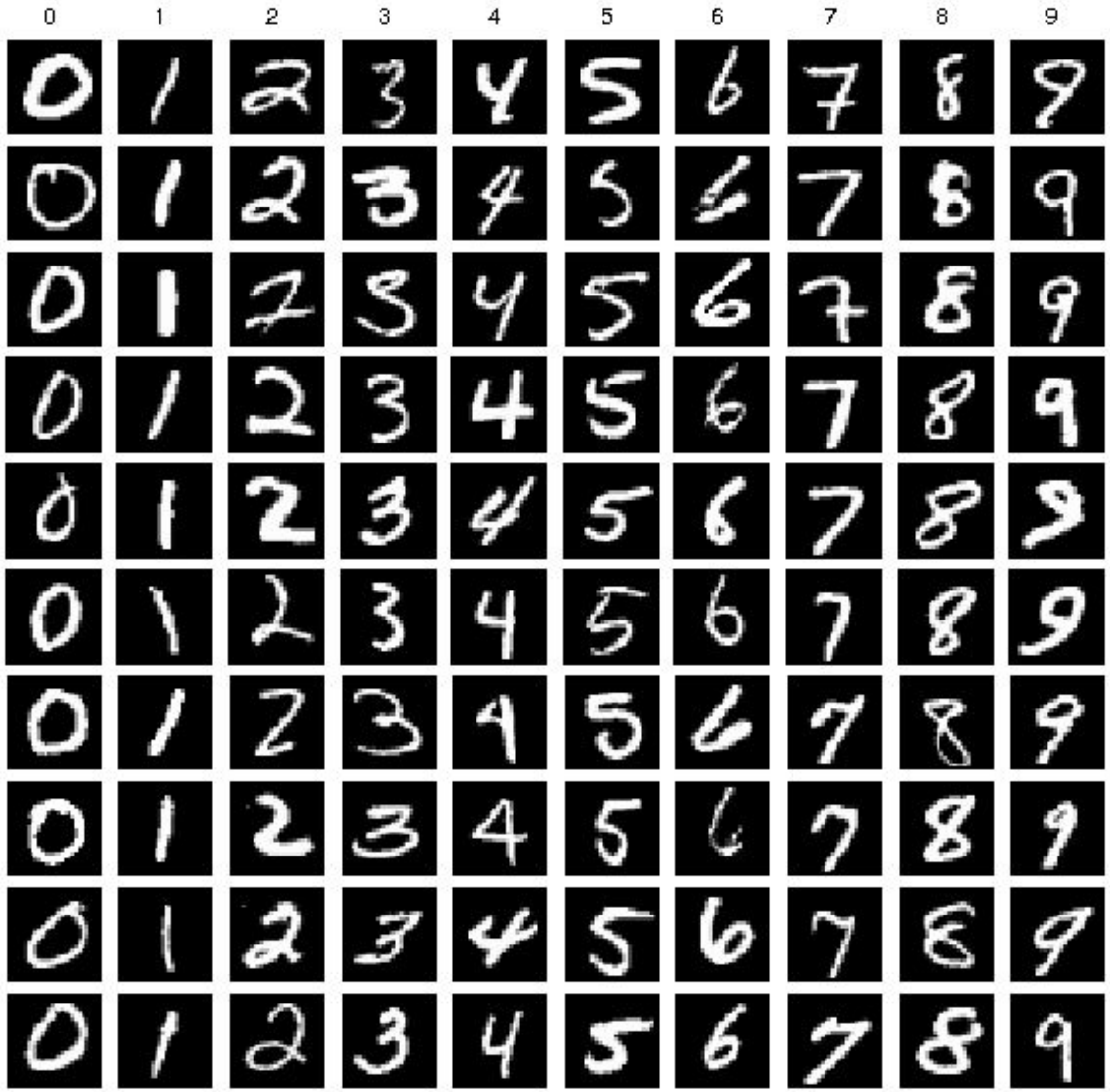
- Max pooling will perform well on sparse data, Boureau et al. 2010[5]
- Average pooling will perform well on dense data
- Pool size can reduce noise in sparse data, same with stride
    - They will both also act as a form of regularization



[2]

Hypotheses (cont.)

- Max pooling will perform well on sparse data, Boureau et al. 2010[5]
- Average pooling will perform well on dense data
- Pool size can reduce noise in sparse data, same with stride
    - They will both also act as a form of regularization



[4]

Experimental Setup

- CNN model built using Keras (see right) [1]
    - Keras is a high-level abstraction of Tensorflow
- Identical models for Max and Average pooling
    - MaxPooling2D -> AveragePooling2D

```python
def compile_CNN(input_shape, num_classes, pool_shape, pool_stride):
    model = Sequential()
    model.add(Conv2D(64, (3, 3), padding='same', input_shape=input_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=pool_shape, strides=pool_stride, padding='same'))
    model.add(Dropout(0.25))

    model.add(Conv2D(64, (3, 3), padding='same'))#, input_shape=input_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=pool_shape, strides=pool_stride, padding='same'))
    model.add(Dropout(0.25))

    model.add(Conv2D(64, (3, 3), padding='same'))#, input_shape=input_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=pool_shape, strides=pool_stride, padding='same'))
    model.add(Dropout(0.25))

    model.add(Flatten())
    model.add(Dropout(0.25))
    model.add(Dense(num_classes))
    model.add(Activation('softmax'))
```

Experimental Setup

Keras CNNs trained on two separate image sets:

CIFAR-10: CIFAR-10 consists of 60,000 32x32 pixel colored images across 10 different classes

- Trained 100 Epochs to reach convergence

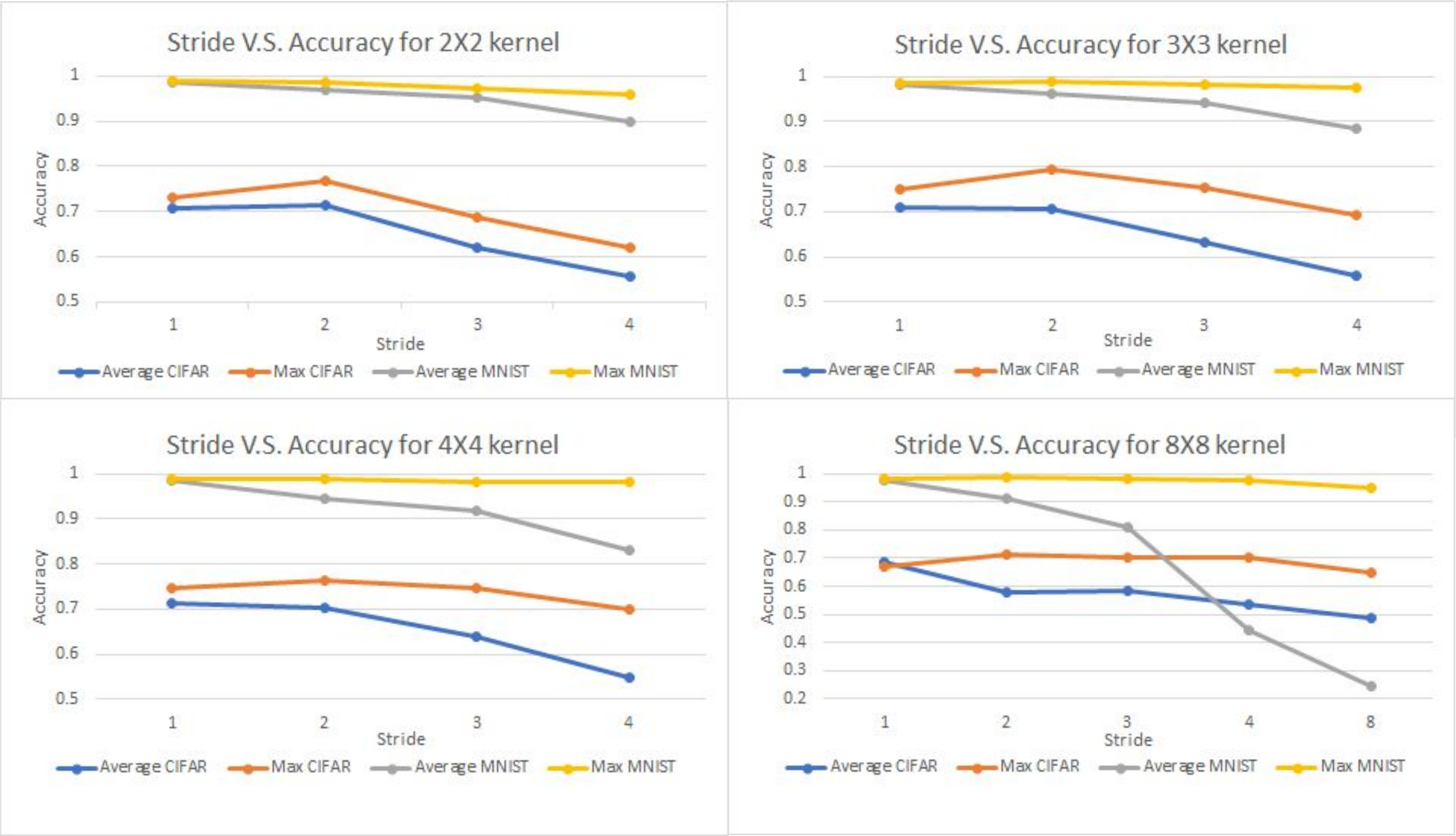
MNIST: MNIST is a database of 60,000 28x28 pixel images of handwritten digits (0 through 9)

- Trained 10 Epochs before convergence

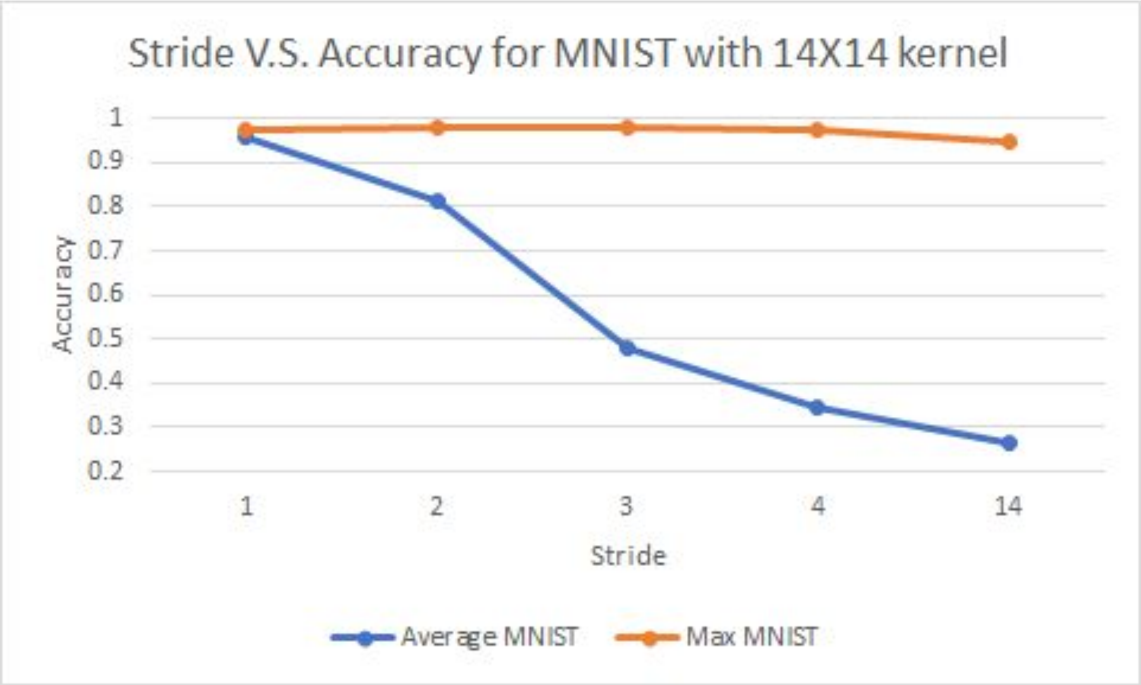
Methodology

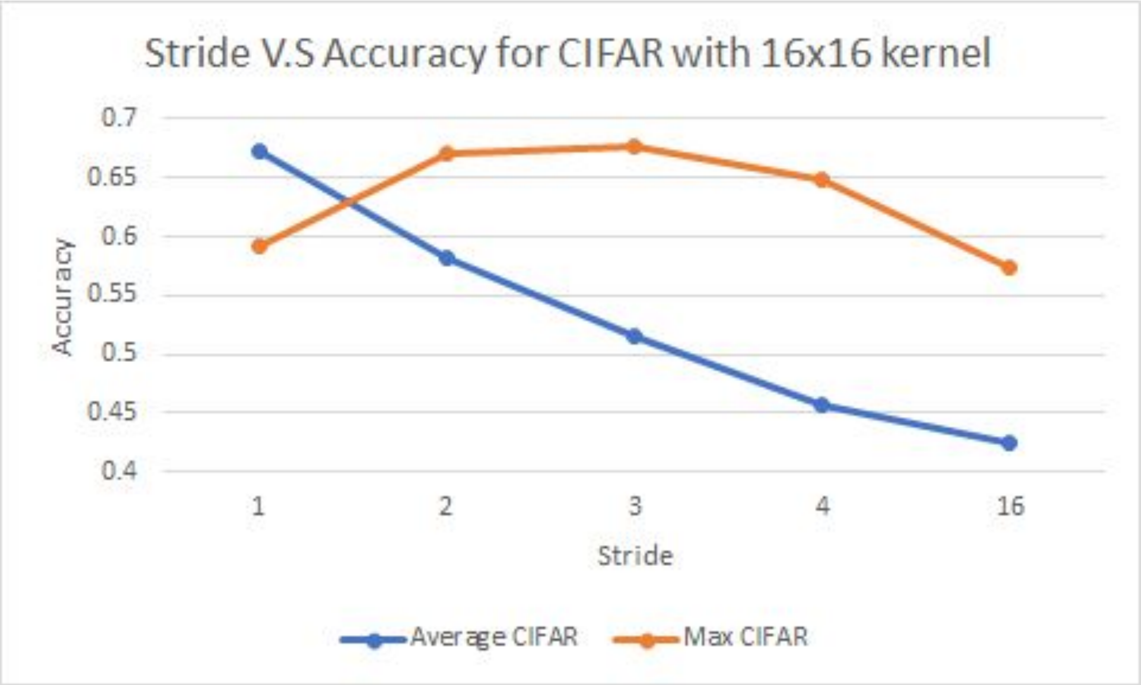
1. Trained 4 (nearly) identical categories of Keras models
   - One model per dataset per pooling type
2. Categories broken into specific models by stride and pool size
   - CIFAR: Pool kernel sizes were 2x2,3x3,4x4,8x8, and 16x16
   - Stride: 1,2,3, and 4 for all
     - 8 for 8x8 and 16 for 16x16
   - MNIST was similar, except 16x16 because 14x14
     - MNIST images are 28x28 instead of 32x32
3. Trained the models on Google's Colaboratory GPU service
4. Collected training, validation, and test accuracy for analysis

Results: Effects of Stride

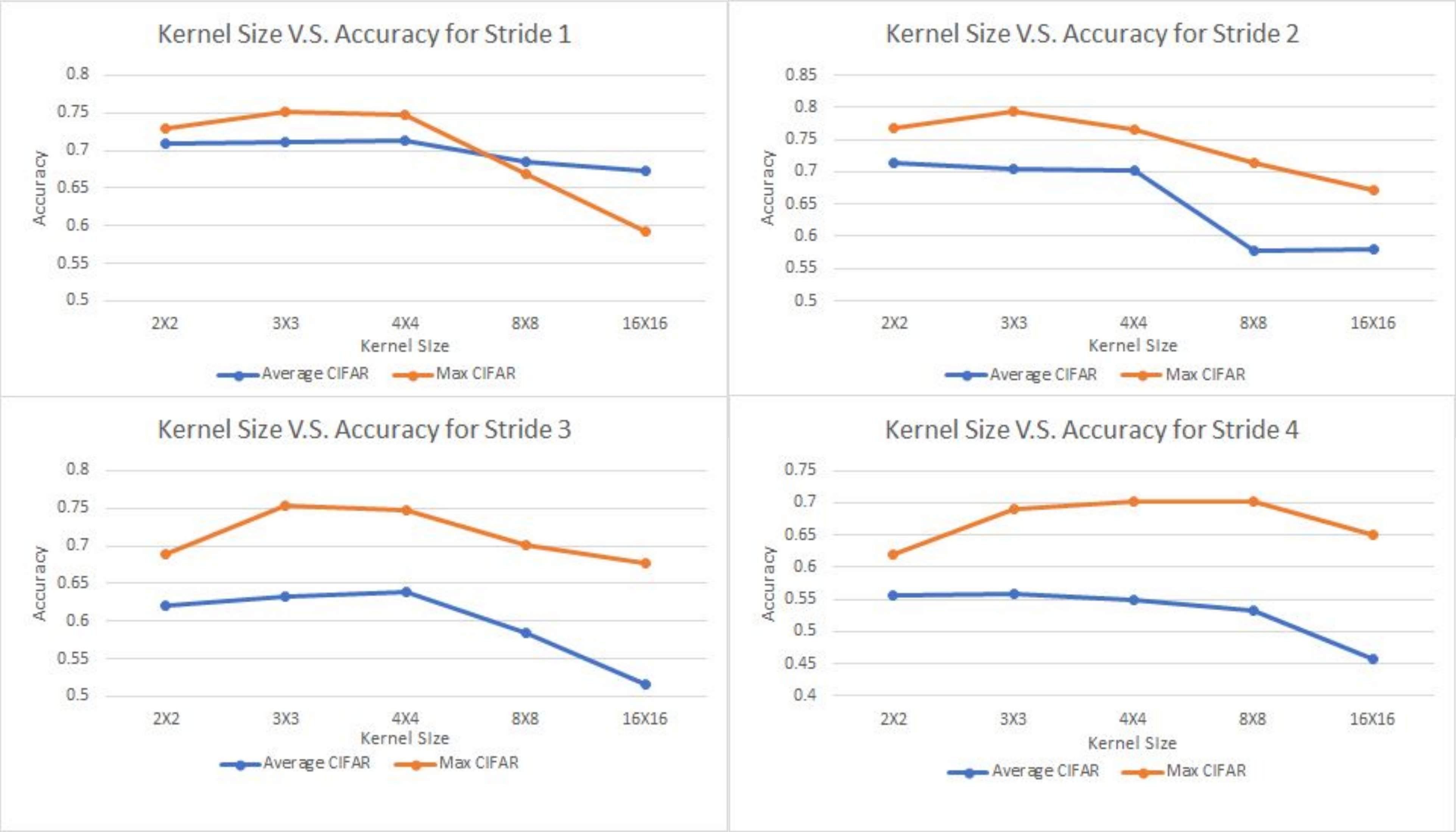


Results: Effects of Stride

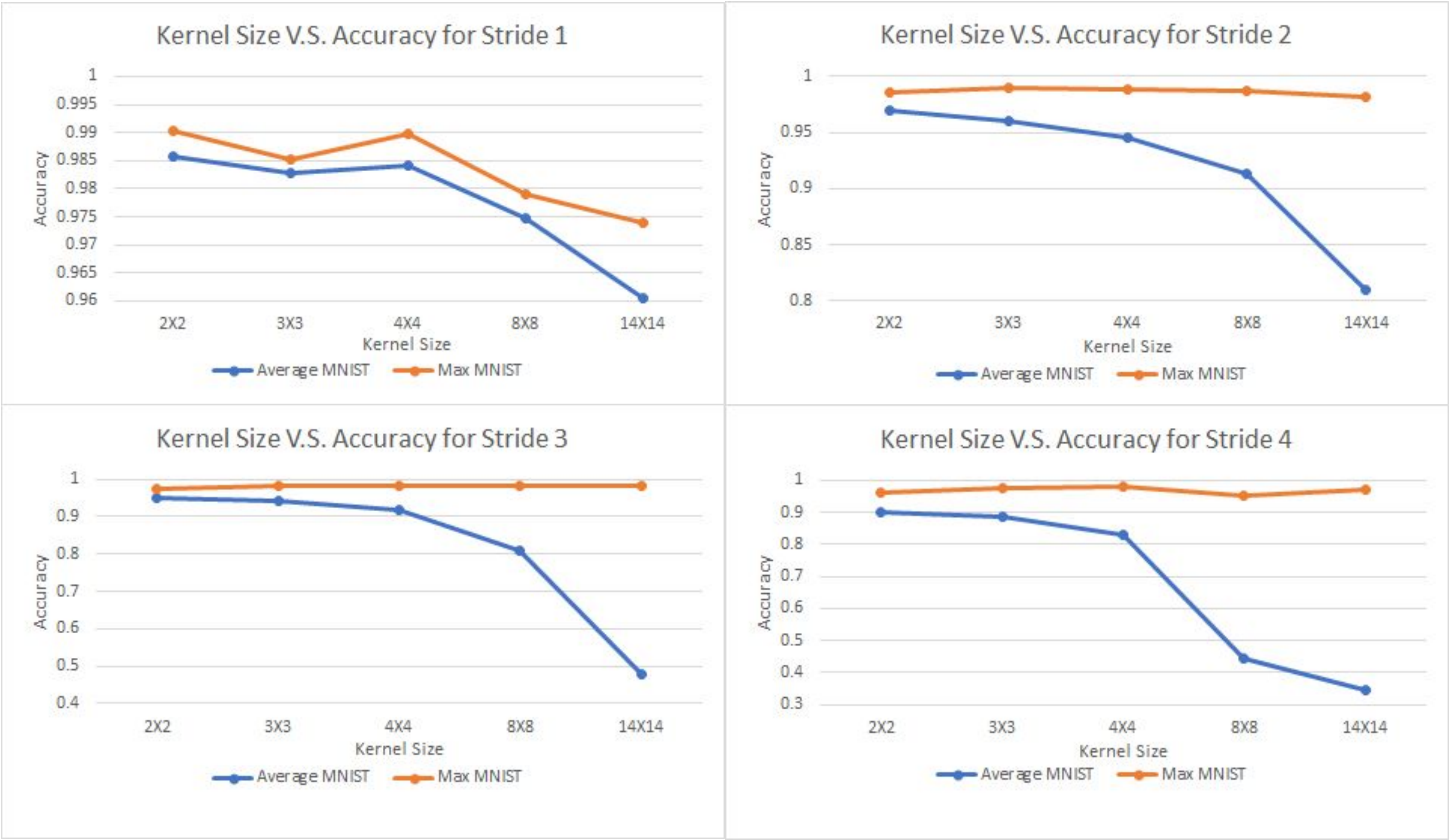




Results: Effects of Pooling Size (CIFAR)



Results: Effects of Pooling Size (MNIST)



Discussion: Max vs. Average

- Max pooling almost always outperforms average pooling
- Max pooling is a non-linear operation
- Average pooling is linear
  - Model capacity may be too low with average alone
  - May succeed in more complicated models, or mixed with max pooling

Discussion: Pool Size and Stride

- Making them too large invariably hurt accuracy
  - Images are very small though
- Pool sizes of 3x3 - 4x4 (about 10-15% of the image size) performed best
- Smaller strides (1-3) also performed the best
  - Higher strides lose a lot of data
  - Strides larger than the pool size are generally harmful

Discussion: Hypotheses

- Max pooling and sparse data
  - Max pooling always outperformed Average on MNIST
  - MNIST is almost all black small white lines
  - CIFAR-10 is very color-dense
- Average pooling and dense data
  - Average pooling occasionally outperformed on CIFAR-10
  - Large kernels and small strides helped
    - Average improves with larger sample area [5]

Discussion: Hypotheses

- Pooling as noise reduction
  - CIFAR-10 could be considered noisy
    - Lots of color, small image
  - Performed quite poorly on CIFAR-10 in general
    - It may not reduce noise well, or our model might be weak
- Pooling as regularization
  - Test accuracy suffered with small stride and large pooling size
  - Improved when stride was also increased
    - Suggests that it can act as regularization
    - Each data point is used less

Discussion: Limitations

- Model seems to underfit CIFAR-10 dataset
  - Originally overfit until Dropout was added
- MNIST gets fairly good results
  - Makes comparison across them difficult
- Could have tested a greater range of sizes/strides
- These datasets are specifically small-image datasets

Conclusions: Summary

- Max outperformed average on “sparse” data
  - Filters out empty space rather than diluting
- Average sometimes outperforms max on dense data
  - Doesn’t lose as much information
  - Happend with large kernels and small strides
- Pooling as a noise filter was inconclusive
- Stride and pool size acted as regularizers
  - Increasing stride decreases the amount each data is used

Conclusions: Recommendations

- Max pooling almost invariably outperforms average pooling
- Uses of average pooling seem very limited
  - Large pooling size, small stride, and you cannot let information be lost
  - Also need extra non-linearities elsewhere
- Strides and pool sizes should generally be kept small
  - Could potentially have a reason to want pool large quadrants
  - 10-15% of image size worked well here

Questions?

References

1. https://keras.io/
2. https://www.cs.toronto.edu/~kriz/cifar.html
3. https://www.kaggle.com/amarjeet007/visualize-cnn-with-keras
4. https://www.researchgate.net/figure/Example-images-from-the-MNIST-dataset_fig1_306056875
5. https://www.di.ens.fr/sierra/pdfs/icml2010b.pdf