Image-Based Plant Leaf Disease Recognition with InceptionV3 Network

Name: Zhengqi Dong

Advisor: Dr. Darren Drewry

03/16/2021

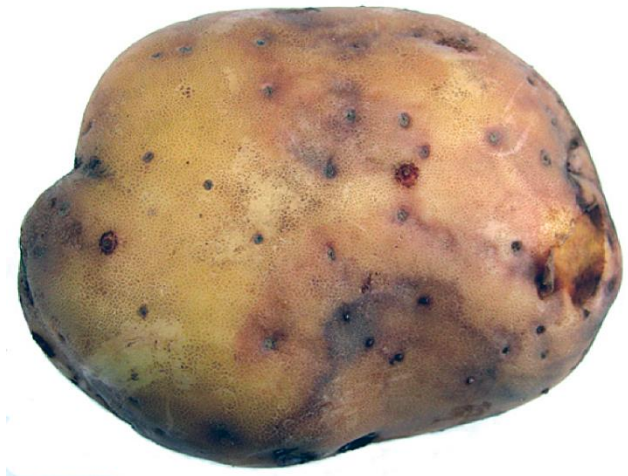
The Ohio State University

Introduction: Plant Diseases

Background of Crop Disease

THE OHIO STATE UNIVERSITY

- History of plant disease outbreak
  - Late blight of potatoes
    - Firstly discovered in early 1840s in Ireland
    - Cause 1 million people died from starvation
    - 2 million people emigrated to other countries.
- Current situation: A huge threaten to food security and food supply world-wide
  - FAO UN: “Must increase food supply by 70% in order to feed 9 billion people by 2050.”
  - Globally, 800 million people don’t have adequate food, and an average of 40% food production yield is lost to infectious disease.
  - In United State, approximately $40 billion of crop yield losses are caused by plant disease annually.
  - In many developing countries, more than 80% of agricultural production is generated by smallholder farmers, and they are particularly vulnerable to pathogen-derived disruptions in food supply



Courtesy: https://www.ag.ndsu.edu/publications/crops/late-blight-in-potato, https://asianlite.com/news/africa-news/un-warns-of-new-wave-of-desert-locusts-in-somalia/

Problem and New Approach

THE OHIO STATE UNIVERSITY

- Problem of human visual based diagnosis
  - Tends to be expensive and time-consuming
  - Not accurate by human visual inspection
  - Can be biased by human previous experiences
  - Not practical for large-scales diagnosis or monitoring
  - Not pragmatic to smallholder farmers
- New Approach: DL integrated smartphone diagnosis app
  - The widespread distribution of smartphone: 7.7 billion smartphone users global wide
  - The ever-growing Smartphone computing power (e.g. iPhone SE):
    - A13 Bionic chip
    - 3rd Neural Engine
    - HD display
    - Ultra wide and 4K camera
  - Light weight novel DL architecture: SqueezeNet, MobileNet, EfficientNet

Figure: iPhone SE Specification

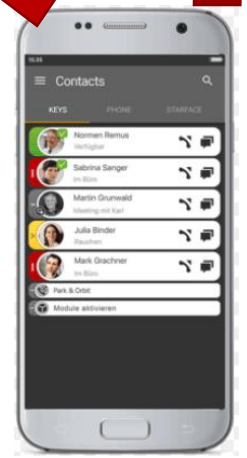
Courtesy: https://www.oberlo.com/statistics/how-many-people-have-smartphones

Application Framework Design

THE OHIO STATE UNIVERSITY

1. Take a picture of infected leaves

2. Preprocessing the images in the phone(e.g. resizing, denoise)



3. Uploading the image to server

4. Evaluate the image and identify the target disease

5. Extract the required information from database

6. Sending the result and suggested treatment to the disease

7. Displaying the analyzed result to client, to help them to make decision.

Courtesy: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-90162018000400273, https://www.ag.ndsu.edu/publications/crops/late-blight-in-potato, https://www.cleanpng.com/png-database-server-computer-servers-computer-icons-cl-3997062/

Methodologies

What made the DL so popular?

THE OHIO STATE UNIVERSITY

- Three things that set the cornerstone of DL:
  - Ingredient 1: Tremendous amount of dataset:
    - ImageNet dataset(1000 categories, 1.2 million images, >300GB)
    - COC dataset(330K images, 1.5 million object instances, 80 object categories), VOC
  - Ingredient 2: Novel algorithmic breakthrough:
    - ConvNet(1989) ➔ LeNet(1998) ➔ AlexNet(2012) ➔ GoogleNet, InceptionV2, V3, V4(2014-2016) ➔ ResNet(2015) ➔ ResNext(2017) ➔ SE-Net, NASNet(2018)
  - Ingredient 3: Computing Power:
    - CPUs:

Core i7 Extreme	Bloomfield	4	45 nm	November 2008				
	Gulftown	6	32 nm	March 2010				
	Sandy Bridge-E	6	32 nm	November 2011	Clarksfield	4	45 nm	September 2009
	Ivy Bridge-E	6	22 nm	September 2013	Sandy Bridge	4	32 nm	January 2011
	Haswell-E	8	22 nm	August 2014	Ivy Bridge	4	22 nm	May 2012
	Broadwell-E	10	14 nm	May 2016	Haswell	4	22 nm	June 2013
	Skylake-X	6-8	14 nm	June 2017				
	Kaby Lake-X	4	14 nm	June 2017				
Core i9	Skylake-X	10	14 nm	June 2017				
	Skylake-X	12	14 nm	August 2017				
	Cascade Lake-X	14-18	14 nm	September 2017	Coffee Lake-H	6	14 nm	April 2018
	Coffee Lake	8	14 nm	October 2018	Comet Lake-H	8	14 nm	April 2020

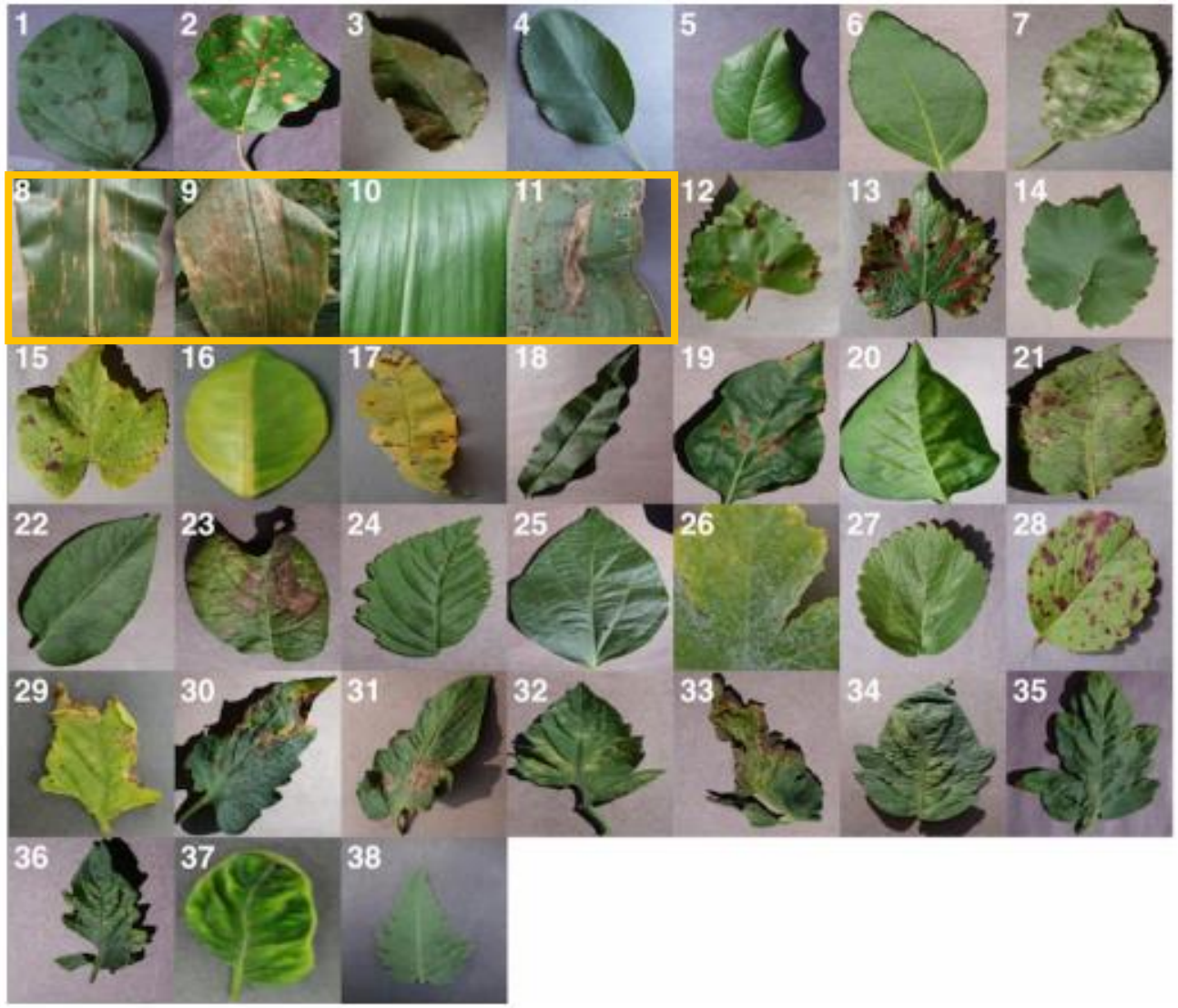
    - GPUs: provides thousands of times more computing cores than CPUs

NVIDIA Card 3000 Series	Number of CUDA Cores	Size of Power Supply **	Memory Type	Memory Interface Width	Memory Bandwidth GB/sec	Base Clock Speed	Boost Clock Speed	NOTES
RTX-3060	3584	550 watt	GDDR6	192 bit	384 GB/s	1320 MHz	1780 MHz	Standard with 12 GB of Memory
RTX-3060 Ti	4864	600 watt	GDDR6	256 bit	448 GB/s	1410 MHz	1670 MHz	Standard with 8 GB of Memory
RTX-3070	5888	650 watt	GDDR6	256 bit	448 GB/s	1500 MHz	1730 MHz	Standard with 8 GB of Memory
RTX-3080	8704	750 watt	GDDR6X	320 bit	760 GB/s	1440 MHz	1710 MHz	Standard with 10 GB of Memory
RTX-3090	10496	750 watt	GDDR6X	384 bit	936 GB/s	1400 MHz	1700 MHz	Standard with 24 GB of Memory

Ingredient 1: Dataset – PlantVilllage Dataset

THE OHIO STATE UNIVERSITY

- PlantVilllage Dataset
  - 54,305 images in total
  - Image size: (256, 256, 3)
  - Classes: 38 categories (26 disease in 14 crop species)
- Sample Corn Disease Dataset
  - Corn Common Rust – 1192 images
    - Caused by fungus Puccinia sorghi, has orange round pustules scattered on surface
  - Corn Gray Leaf spot – 513 images
    - caused by the fungus Cercospora zeae-maydis, has narrow, rectangular, and light tan-colored lesions
  - Corn Healthy – 1162 images
  - Northern Corn Leaf blight – 985 images
    - caused by the fungus Setosphaeria turcica, has elliptical, and pale gray or tan colored lesions.
- Result for Corn Disease Dataset:
  - Train Acc: 99.5%, Valid Acc: 97.6%



Courtesy: https://arxiv.org/ftp/arxiv/papers/1604/1604.03169.pdf

Ingredient 2: Novel Algorithm—InceptionV3

THE OHIO STATE UNIVERSITY

- **Background:**
  - Published in 2015 by Christian Szegedy, and etc, with 12575 citations.
  - Third edition of Google's Inception Convolutional Neural Network: GoogLeNet(InceptionV1) → BN-Inception(InceptionV2)→ InceptionV3
- **Innovative Ideas:**
  - **Factorized Convolutions**: use factorized convolution module to reduce the number of parameter and provides a higher computational efficiency.
  - **Auxiliary Classifier:** Use auxiliary classifier as regularizer to combat the vanishing gradient problem in very deep network.
  - **Efficient Grid Size Reduction:** Instead of using the pooling operation to downsize the feature map, a variant approach is used to reduce the bottleneck of computation cost.

Courtesy: https://arxiv.org/pdf/1512.00567.pdf

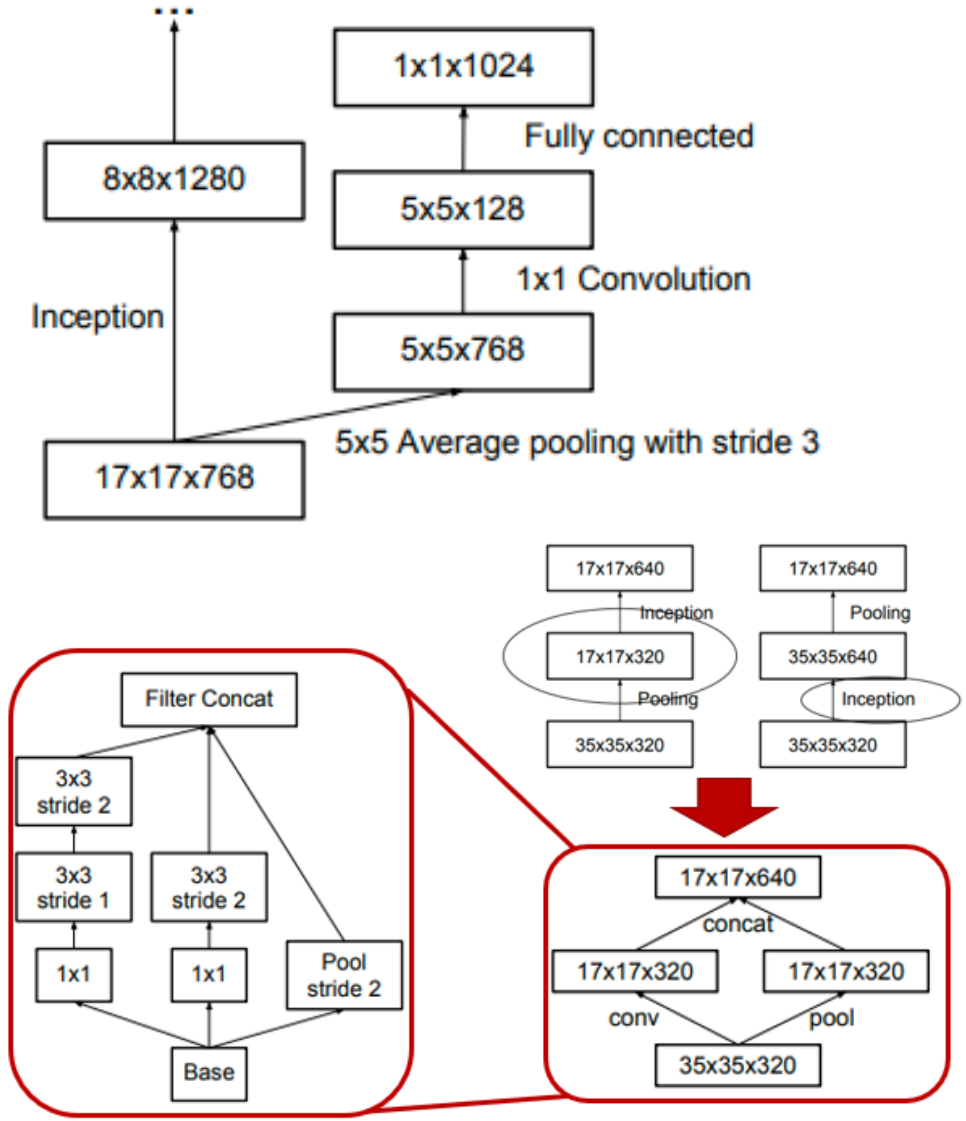
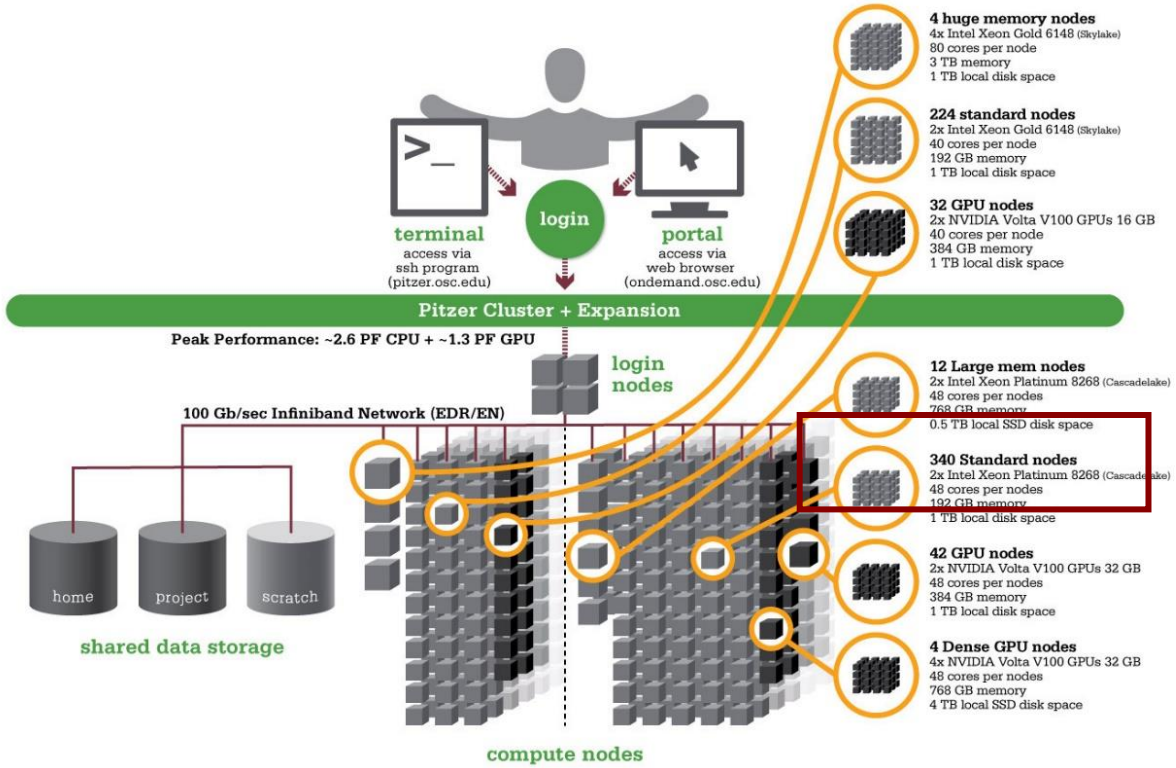


Figure: Auxiliary Classifier(Top), Efficient Grid Size Reduction(Bottom)

Ingredient 3: Computing Power – OSC clusters

THE OHIO STATE UNIVERSITY

- Computing Power:
  - OSU supercomputing center(OSC) Pitzer cluster:
    - Dual NVIDIA Volta V100 GPUs
    - Average performance: ~1300 TF
    - Peak Performance: ~2.6 PF CPU + ~1.3 PF GPU



Hardware/Software	Configurations
Memory	384GB on CPU and 32GB on GPU
Disk	1TB
Computing Processing Unit(CPUs)	Dual Intel Xeon Platinum 8268s Cascade Lakes@2.9GHz x 48 cores
Graphics Processing Unit(GPUs)	Dual NVIDIA Volta V100
Operating System	Red Hat Enterprise Linux Server release 7.7
Python	3.6.5
TensorFlow	2.4.0
CUDA	11.0.3

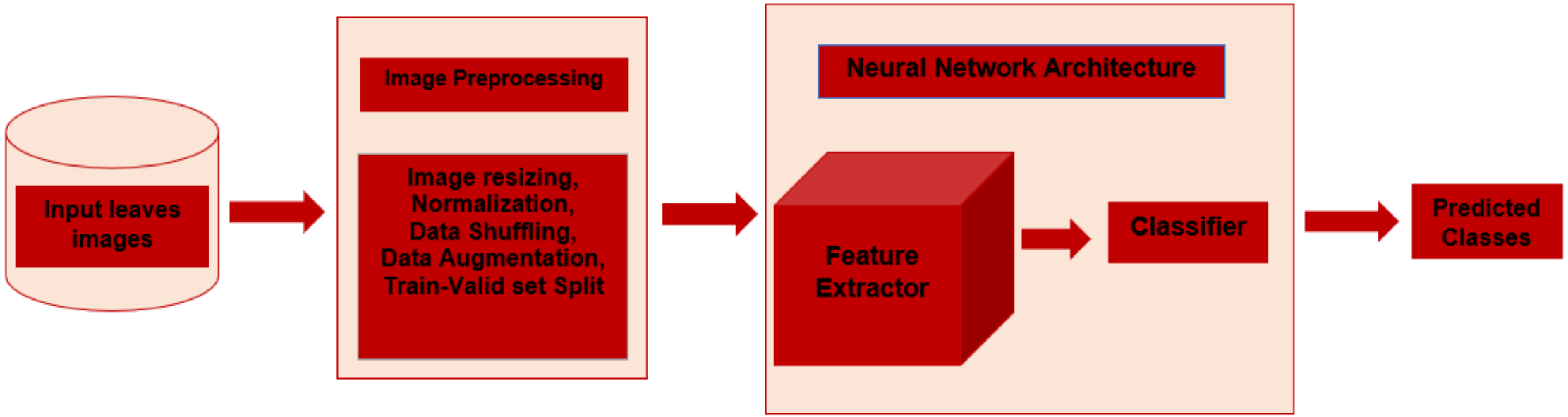
Table: Hardware configuration support for this project

Courtesy: https://www.osc.edu/resources/technical_support/supercomputers/pitzer

Experimental Design:

THE OHIO STATE UNIVERSITY

- Applied techniques for data preprocessing:
  - Image Resizing: standardized the image size, (?, ?) → (256, 256)
  - Normalization: (0, 255) → (0,1)
  - Shuffling dataset before training
  - Data augmentation:
    - rotation_range=25, width_shift_range=0.1, height_shift_range=0.1, height_shift_range=0.1, zoom_range=0.2, fill_mode='nearest', horizontal_flip=True
  - Train-Test Split: 0.1 test split ratio
    - Training Dataset: The sample of data used to fit the model
    - Validation Dataset: The sample of data used to tune the parameters during training, e.g., evaluate the result after each epoch.
    - Testing Dataset: The sample of data used to provide an unbiased evaluation of a final model



Result and Analysis

Evaluation Metrics

THE OHIO STATE UNIVERSITY

Description:

- **Precision:**
  - Measure the percentage of correct positive prediction. E.g., A model that produce no false positive has precision of 1.0.
- **Recall:**
  - Measure the percentage of actual positive prediction. E.g., a model that produce no false negative has a recall of 1.0.,
- **F1 Scores:**
  - F1 score is a weighted harmonic mean of recall and precision
- **AUC:**
  - Stands for "Area under the ROC Curve“, measure the entire two-dimensional area underneath the entire ROC curve (think the integral of ROC curve) from (0,0) to (1,1).
- **Accuracy:**
  - Measure the number of correct predictions that had been made over the data.
- **Loss Function:**
  - **Categorical Cross Entropy:** A loss function that mostly used in multi-class classification and apply the one-hot encoding to compute the target score.

Mathematical Expression

- **Precision:**
  - $Precision = \frac{TP}{TP+FP} = \frac{TP}{Total\ positive\ result}$
- **Recall:**
  - $Recall = \frac{TP}{TP+FN} = \frac{TP}{All\ result}$
- **F1 Scores:**
  - $F1 = 2 * \frac{Precision*Recall}{Precision+Recall}$
- **Accuracy:**
  - $Accuracy = \frac{\#\ of\ corrected\ prediction}{Total\ Samples}$
- **Categorical Cross Entropy:**
  - Cross−Entropy(CE) $= -\sum_i^C \boldsymbol{y}_i log\ (f_i(\boldsymbol{x_i}; \boldsymbol{\theta}))$

Train-Validation Split Ratio

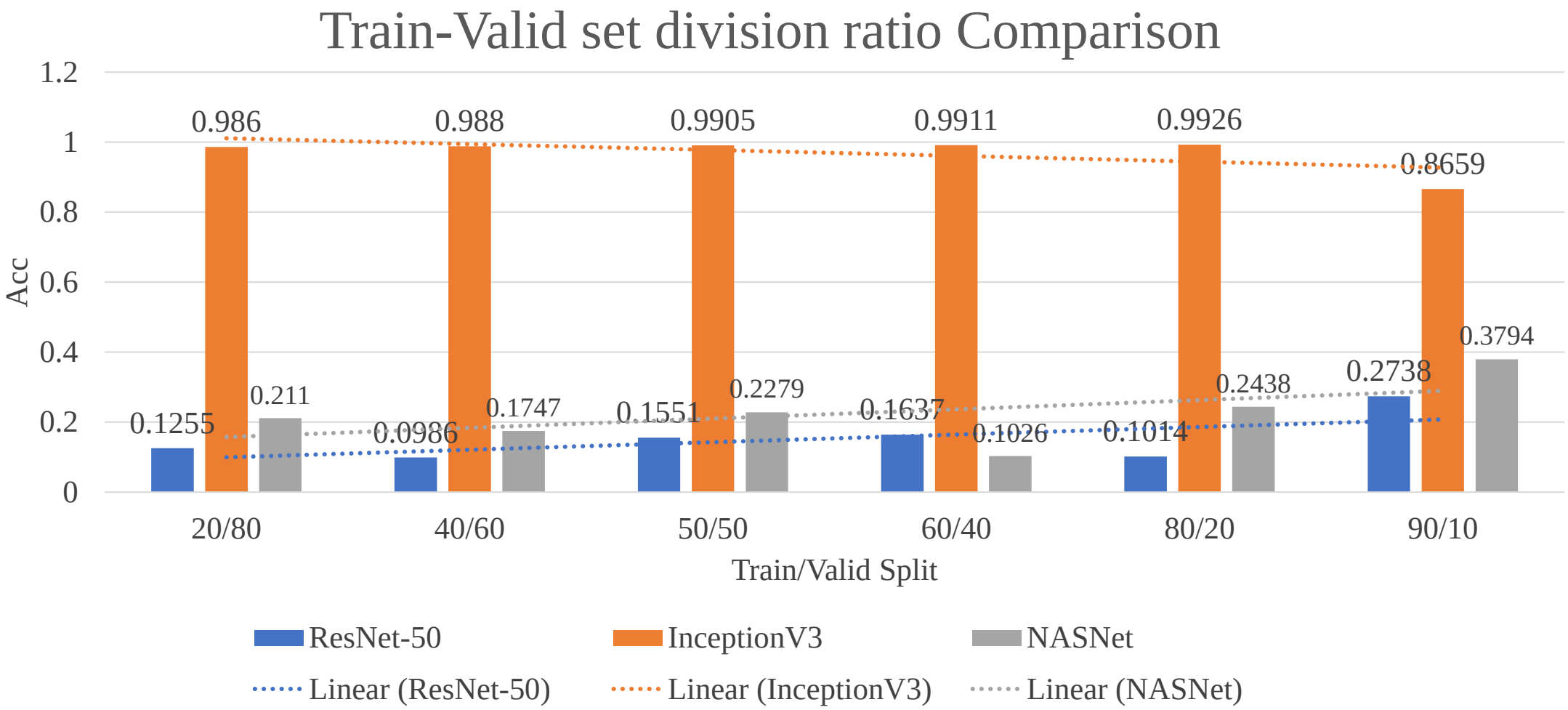


Figure: The training ACC for different train-validation split ratio on 3 different DNN models after 3 epochs of training

Batch Size Comparison on InceptionV3 Network

THE OHIO STATE UNIVERSITY

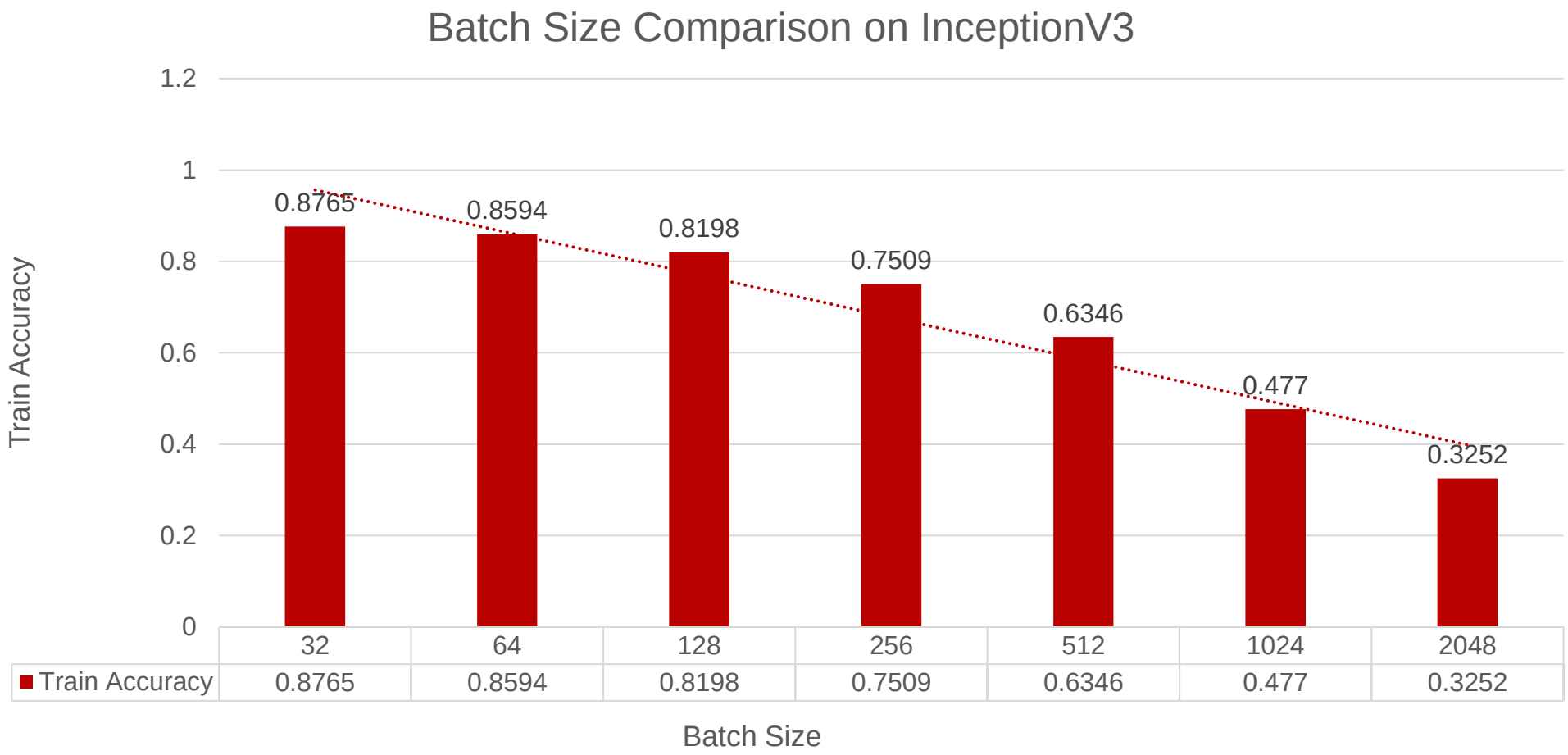


Table: The performance evaluation of various batch size on InceptionV3 network.

CNN Models Performance Evaluation and Analysis

THE OHIO STATE UNIVERSITY

Model	Train accuracy %	Validation accuracy %	Training loss	Validation loss	Precision%	Recall %	AUC%	Valid precision %	Valid recall %	Valid AUC %
ResNet-50	17.68	15.90	2.9988	3.1075	48.16	2.05	79.51	0.00	0.00	77.80
InceptionV3	97.94	96.04	0.0587	0.1237	98.15	97.77	99.95	96.41	95.81	99.78
NASNetMobile	48.27	47.68	1.7768	1.8007	78.07	32.87	93.87	81.57	30.08	93.78
NASNetLarge	9.27	9.87	3.3834	3.3714	0.00	0.00	69.08	0.00	0.00	69.20
MobileNet-v3-small	17.36	17.23	2.9898	2.9602	55.00	0.72	79.90	24.13	0.06	80.64
MobileNet-v3-large	22.45	20.96	2.8342	2.8540	57.39	1.95	82.81	33.33	0.07	82.87

Table: Comparing the performance of 6 different CNN models after 40 epochs of training.

Note: The above experiments was conducted on Pitzer with 48 cores(maximum we can allocated) GPUs, and the result was measured after 1 epochs of training. Also, ResNet50 and InceptionV3 applied the weight initialization of ImageNet, and the NASNet is trained from scratch.

Model Evaluation: Model and Validation Accuracy Comparison

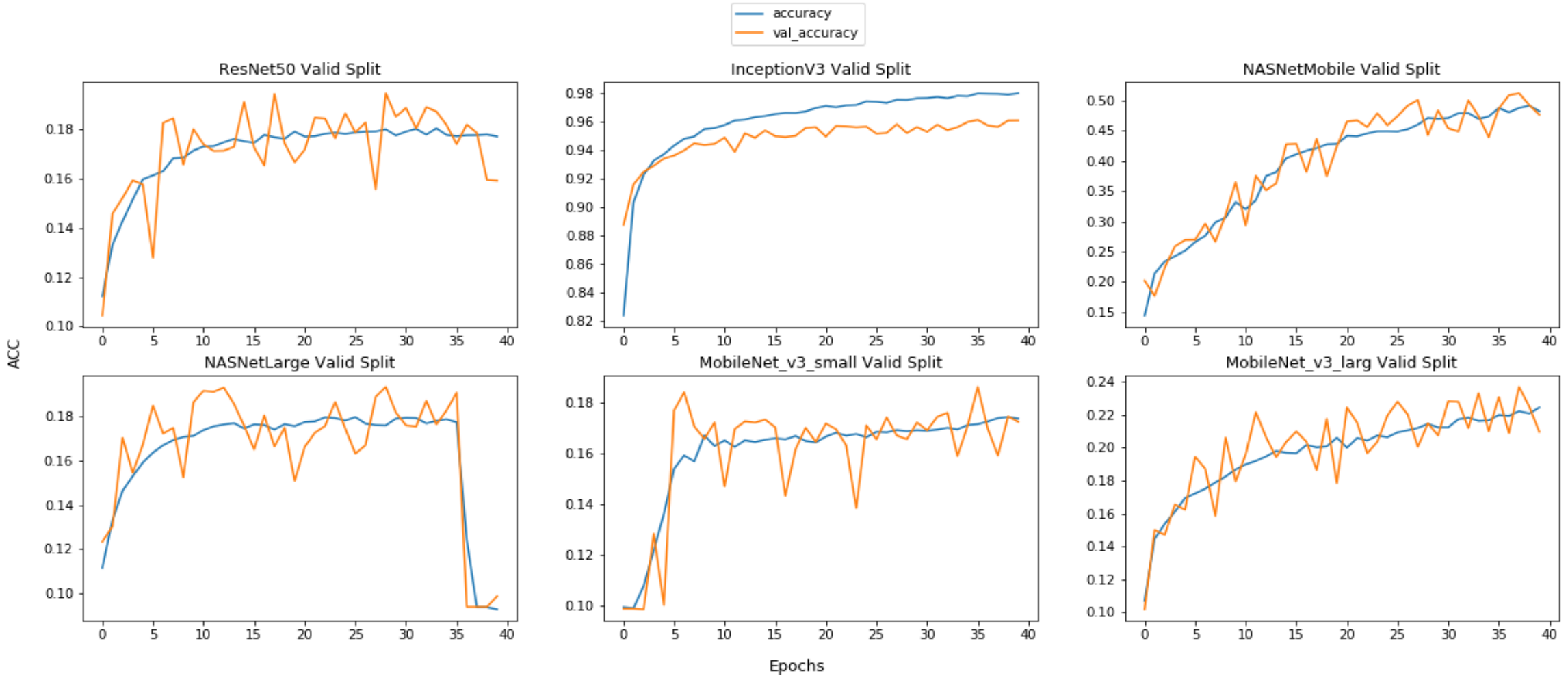


Figure: Comparing the training and validation accuracy of 6 different CNN models after 40 epochs of training.

Model Evaluation: Model and Validation loss Comparison

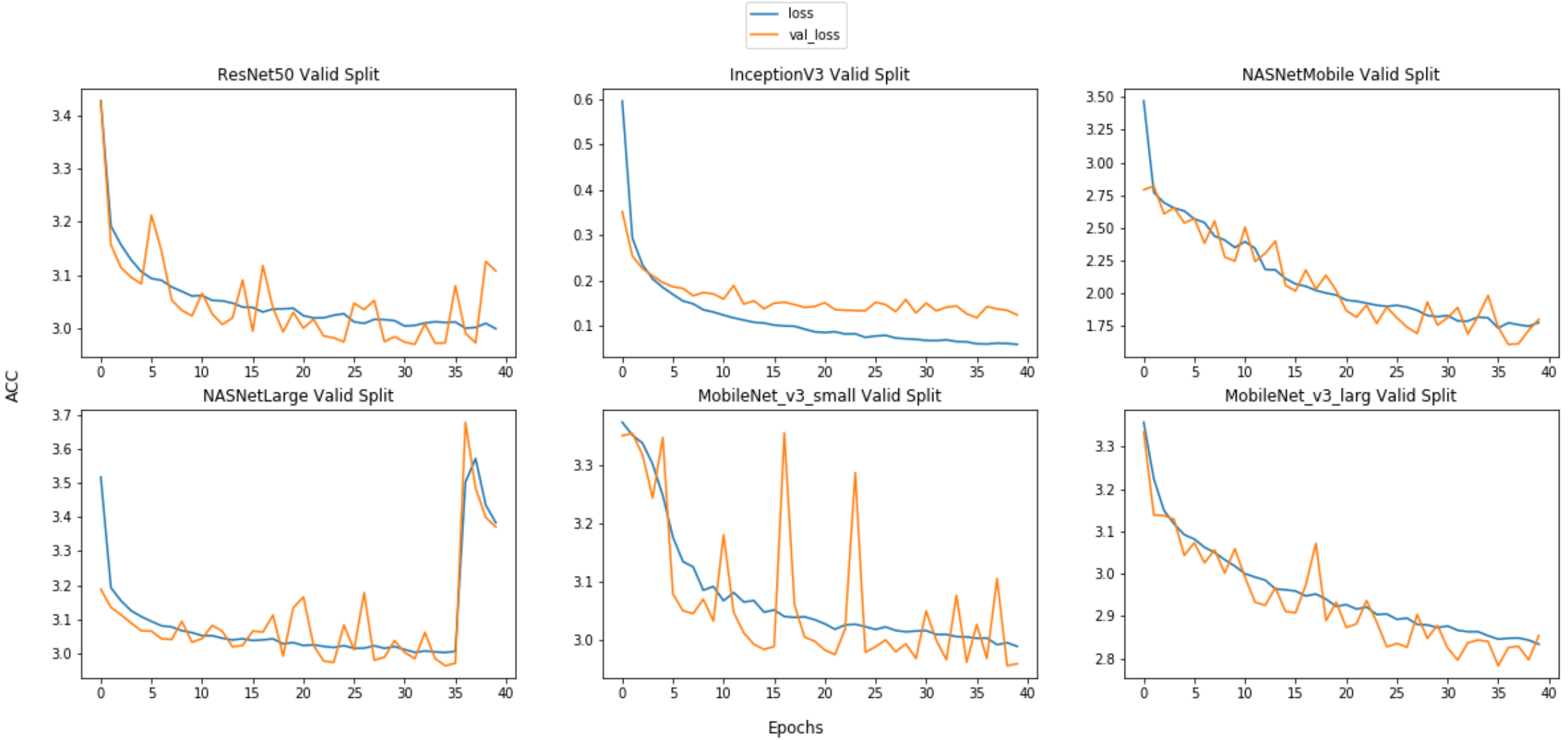


Figure: Comparing the training and validation loss of 6 different CNN models after 40 epochs of training.

Conclusion:

THE OHIO STATE UNIVERSITY

- As the training ratio on the dataset increased, the training accuracy will increase. The best train-validation split ratio for our study is 80/20.
- As the batch size increased, the training accuracy for the model will increase. The best batch size for mini-batch gradient descent is 32 for our study.
- By comparing the performance of different size CNN models, no obvious linear correlation has found between the model complexity and their generalization ability.
- With the InceptionV3 neural network, the best performance we achieved after 40 epochs of training was 97.94% and 96.04% on the training and validation dataset, respectively, which outperformed other candidate models by a significant amount.
- Specifically, the InceptionV3 can achieve top-1 accuracy of 94.14% in training and 94.53% in validation over 2-hour of training and 97.94% in training and 96.04% in validation over 16-hours, respectively, on OSC Pitzer cluster with 48 cores of Dual NVIDIA Volta V100 GPUs, which suggests that the InceptionV3 model is suitable for both lightweight smart applications and backend workstation development within the context of plant leaf disease recognition

Thank You!
Any Question?

Reference:

THE OHIO STATE UNIVERSITY

[1] IFAD, UNEP. "Smallholders, food security and the environment." Rome: International Fund for Agricultural Development (2013).

[2] Martinelli, Federico, et al. "Advanced methods of plant disease detection. A review." Agronomy for Sustainable Development 35.1 (2015): 1-25.

[3] Everingham, Mark, et al. "The pascal visual object classes (voc) challenge." International journal of computer vision 88.2 (2010): 303-338.

[4] Mohanty, Sharada P., David P. Hughes, and Marcel Salathé. "Using deep learning for image-based plant disease detection." Frontiers in plant science 7 (2016): 1419.

[5] Fuentes, Alvaro, et al. "A robust deep-learning-ba real-time tomato plant diseases and pests recognition." Sensors 17.9 (2017): 2022.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. 2016 Ieee Conf Comput Vis Pattern Recognit Cvpr (2016), 770–778. DOI:https://doi.org/10.1109/cvpr.2016.90

[7] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. Retrieved August 11, 2020 from https://arxiv.org/pdf/1512.00567.pdf

[8] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2017. Learning Transferable Architectures for Scalable Image Recognition. Arxiv (2017).

Batch Size

THE OHIO STATE UNIVERSITY

Batch size	Train Acc	Val Acc	Time/Epoch
32	0.8765	0.9544	1064
64	0.8594	0.9489	1159
128	0.8198	0.9289	1085
256	0.7509	0.9008	1164
512	0.6346	0.8554	1045
1024	0.4770	0.7300	1176
2048	0.3252	0.5662	1132

Table: Training Accuracy on Various (32/64/128/256/512/1024/2048) Batch Size

Inception-v3 module A: Factorization into smaller convolutions

THE OHIO STATE UNIVERSITY

Original 5x5 convolution in GoogLeNet(Inception-v1)

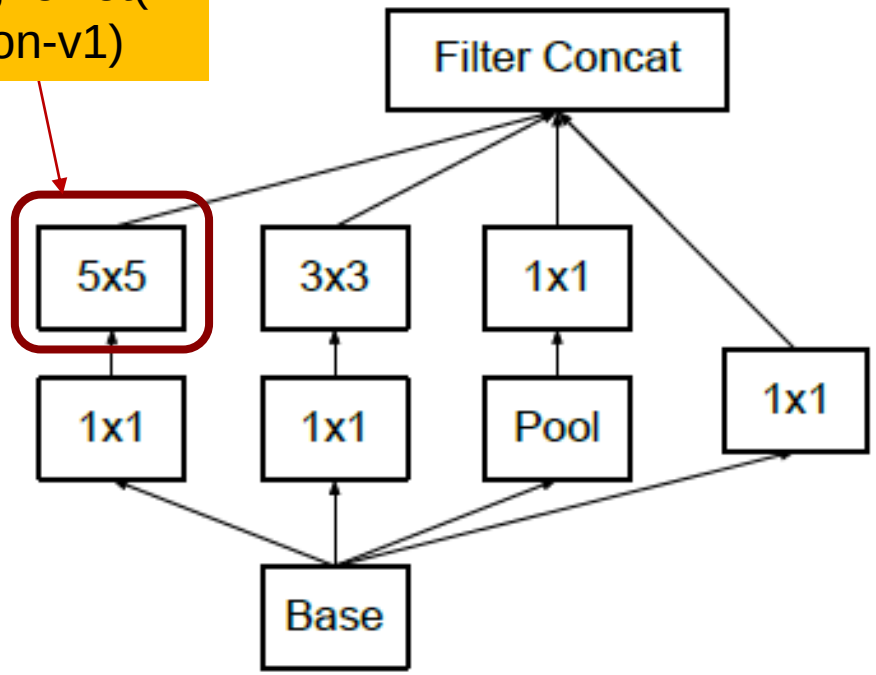


Figure: Original Inception module described in GoogLeNet

Inception modules where 5x5 conv is replaced by two 3x3 conv in reception-v3

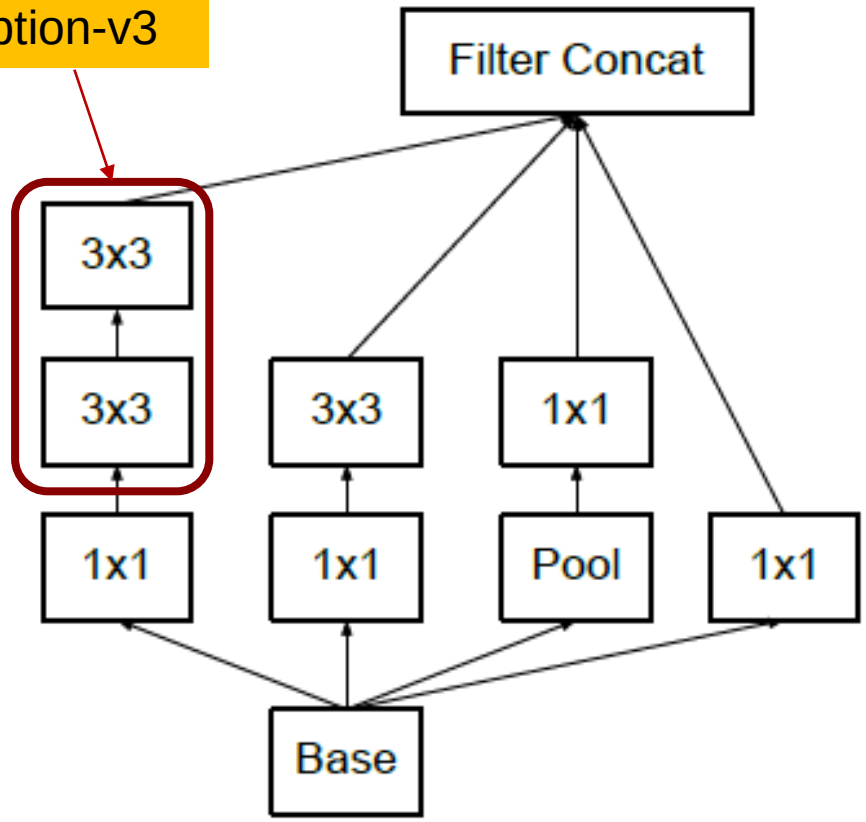


Figure: New Inception module A, where 5x5 conv is replaced by two 3x3 conv in Inception-v3

- With 5x5 conv, number of parameters = 5*5=25
- With two 3x3 conv, number of parameters = 3*3 + 3*3 = 18
- Number of parameters is reduced by (25-18)/25 = 28%

Courtesy: 1)GoogLeNet, https://arxiv.org/abs/1409.4842, 2)Inception-v3, https://arxiv.org/abs/1512.00567

Inception-v3 module B: Factorization into Asymmetric convolutions

Replacing two 7x7 conv with 1x7 and 7x1 conv (where n=7 in implementation)

Replacing one 7x7 conv with 1x7 and 7x1 conv

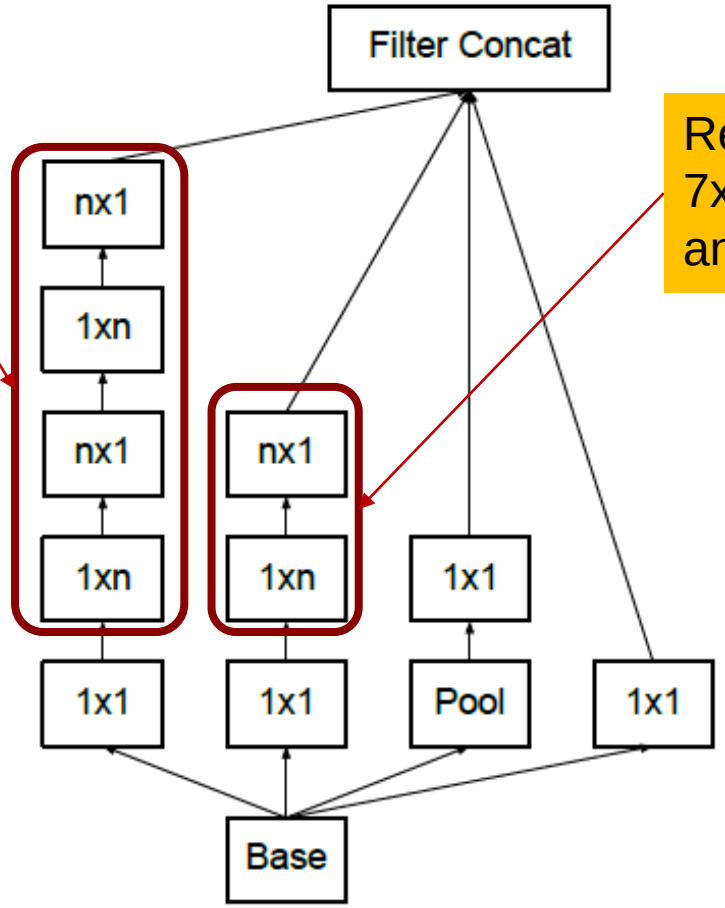


Figure: Inception module B with asymmetric factorization

- With 7x7 conv, number of parameters = 7*7 = 49
- With 1x7 and 7x1 conv, number of parameters = 1*7 + 7*1 = 14
- Number of parameters is reduced by (49-14)/49 = 71.4%

Train-Validation Split Ratio

THE OHIO STATE UNIVERSITY

Train-Validation split\NN model	ResNet-50	InceptionV3	NASNetMobile
Train 20%, Valid 80%	0.1255	0.9860	0.2110
Train 40%, Valid 60%	0.0986	0.9880	0.1747
Train 50%, Valid 50%	0.1551	0.9905	0.2279
Train 60%, Valid 40%	0.1637	0.9911	0.1026
Train 80%, Valid 20%	0.1014	0.9926	0.2438
Train 90%, Valid 10%	0.2738	0.8659	0.3794

Table: Accuracy score across various experiment configurations after 3 epochs of training.

Auxiliary Classifier

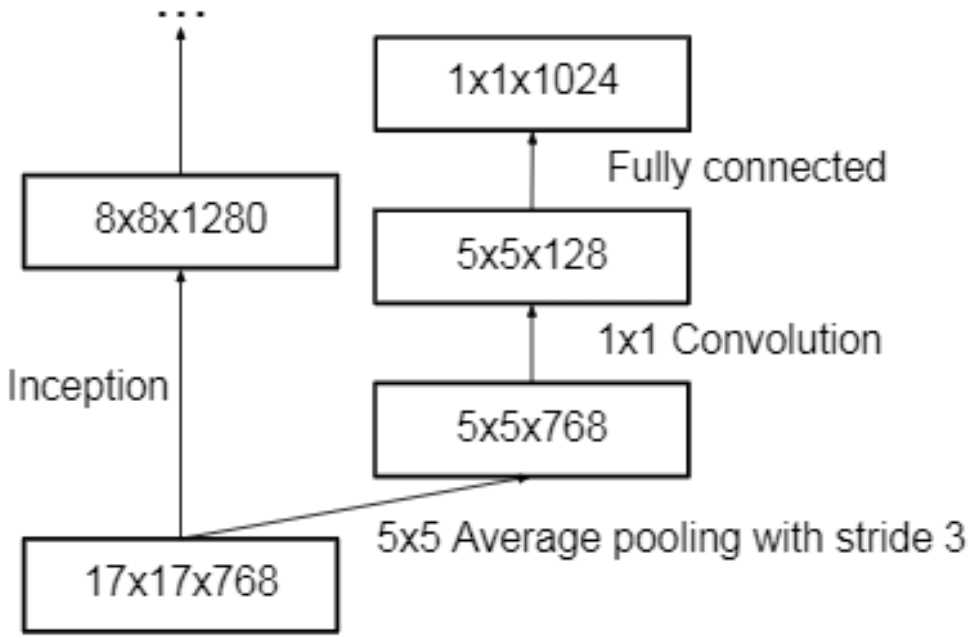


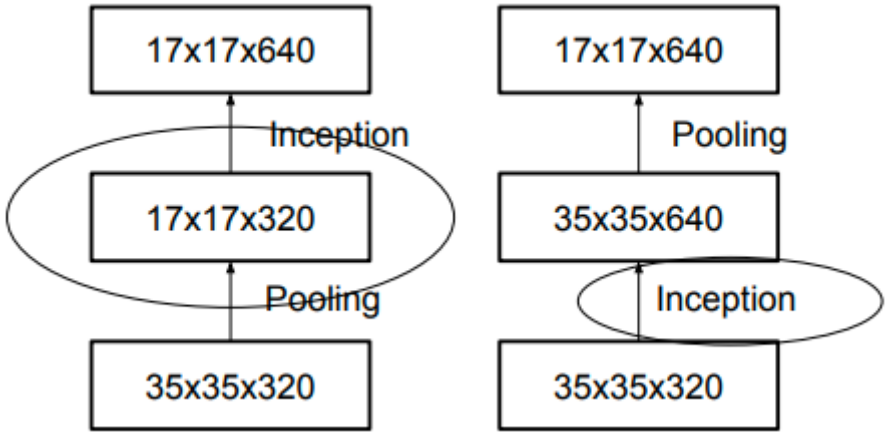
Figure 8. Auxiliary classifier on top of the last 17×17 layer. Batch normalization[7] of the layers in the side head results in a 0.4% absolute gain in top-1 accuracy. The lower axis shows the number of itertions performed, each with batch size 32.

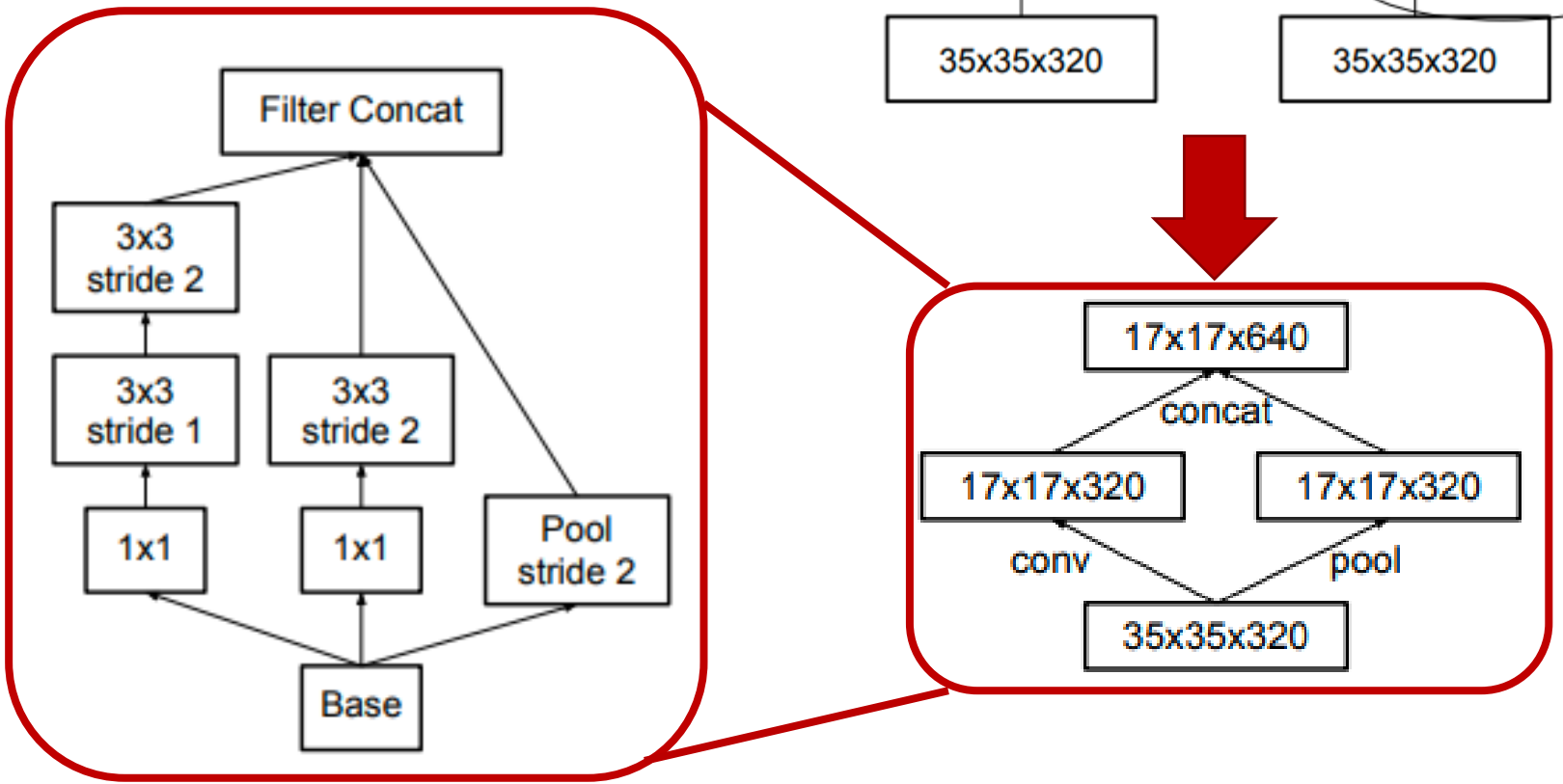
Courtesy: https://arxiv.org/pdf/1512.00567.pdf

Efficient Grid Size Reduction

THE OHIO STATE UNIVERSITY

- Original Approach:
  - Either violates the principle of 1 (no representational bottleneck block introduced) or more computational expensive.
- New Approach:
  - Less computational expensive and still efficient in building very deep neural network.





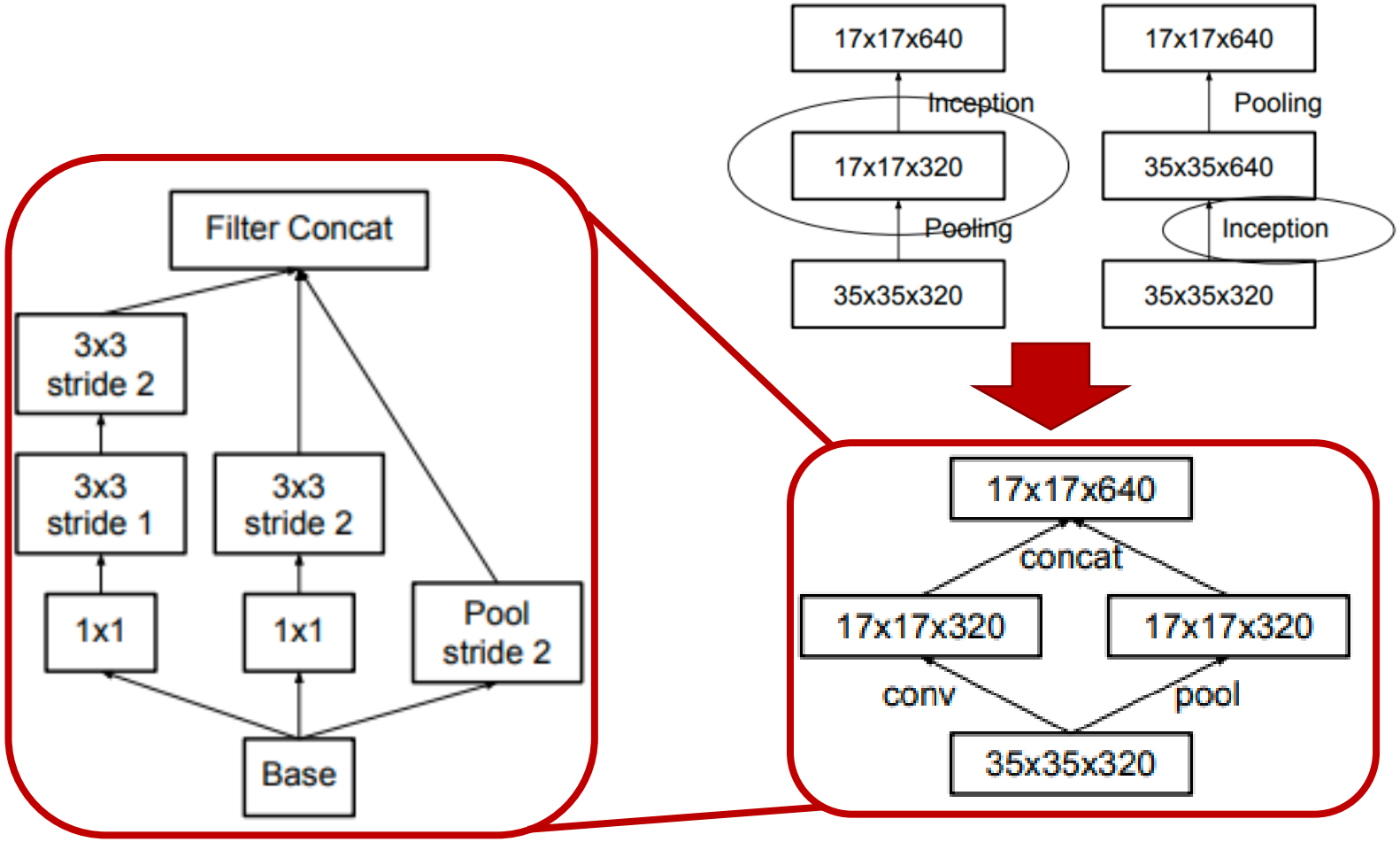


Figure: Original downsizing with pooling layer (Top Right), Efficient Grid Size Reduction (Bottom Right), Detailed Architecture of Efficient Grid Size Reduction (Left)

Experiment: Performance on ResNet101

THE OHIO STATE UNIVERSITY

Model	Training Time (sec/epoch)	Train Acc(%)	Valid Acc(%)	Test Acc(%)
ResNet101(39 epochs)	87~93 sec	64.22%	66.84%	86.49% (16 images)
InceptionV3(50 epochs)	51~66 sec	99.5%	97.6%	93.7%