A Robust Sampling for Optimization in Deep Learning

Master thesis

Aurora Cobo Aguilera

Tutor: Antonio Artés Rodríguez
University Master in Multimedia and Communications
Signal Theory and Communications Department
Universidad Carlos III de Madrid

Leganés, 26th July 2018

uc3m

Introduction

A Robust Sampling for Optimization in Deep Learning

uc3m

Introduction

A Robust Sampling for Optimization in Deep Learning

?

uc3m

Introduction

A **Robust** Sampling for Optimization in Deep Learning

Introduction

A **Robust** Sampling for Optimization in Deep Learning

(Namkoong & Duchi, 2017) → **Regularized estimation based on minimizing the variance**

Introduction

Introduction

A Robust Sampling for **Optimization** in Deep Learning

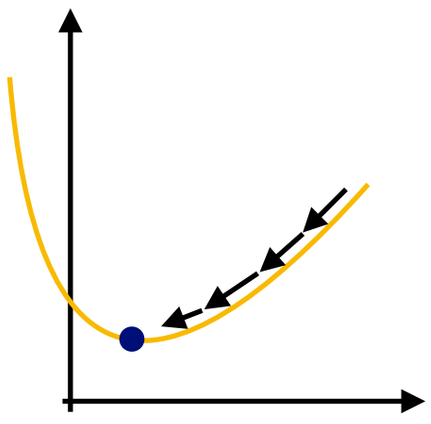
Empirical Risk Minimization

$$\min_{\theta}$$

uc3m

Introduction

A Robust Sampling for Optimization in Deep Learning



Empirical Risk Minimization

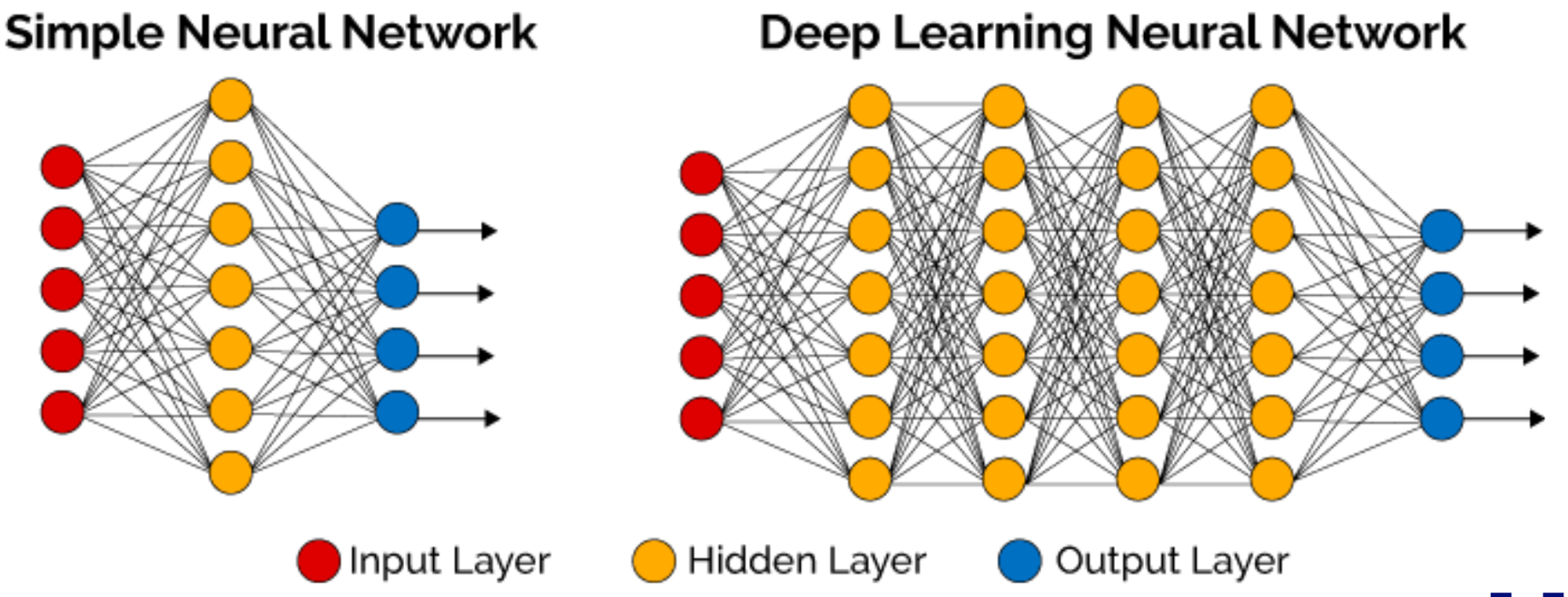
$$\min_{\theta} \; R_{emp}(\theta) = \frac{1}{n} \sum_{i=1}^{n} Q\left(x_i, \theta\right)$$

uc3m

Introduction

Introduction

A Robust Sampling for Optimization in **Deep Learning**



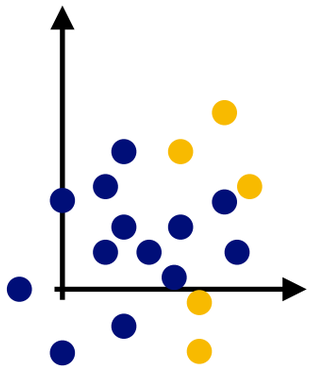
Introduction

A Robust **Sampling** for Optimization in Deep Learning

uc3m

Introduction

A Robust **Sampling** for Optimization in Deep Learning

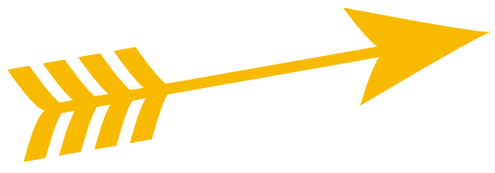


Select the samples of the mini-batch

uc3m

Objectives

1. Improve the performance
2. A faster convergence
3. Better explanation of outliers



Objectives

1. Improve the performance
2. A faster convergence
3. Better explanation of outliers

Supply an application to Deep Learning
Supply the numerical evidence

uc3m

Empirical Risk

What we would like to have

What we have

Empirical Risk

What we would like to have

- Real distribution of the data

What we have

uc3m

Empirical Risk

What we would like to have

- Real distribution of the data
- True risk

What we have

Empirical Risk

What we would like to have

- Real distribution of the data
- True risk

$$R(f) = \mathrm{E}\left\{Q(f(x))\right\}$$

What we have

Empirical Risk

What we would like to have

- Real distribution of the data
- True risk

$$R(f) = \mathrm{E}\left\{Q(f(x))\right\}$$

What we have

- A set of samples $\{x_i, y_i\}$

uc3m

Empirical Risk

What we would like to have

- Real distribution of the data
- True risk

$$R(f) = \mathrm{E}\left\{Q(f(x))\right\}$$

What we have

- A set of samples $\{x_i, y_i\}$
- Empirical risk

Empirical Risk

What we would like to have

- Real distribution of the data
- True risk

$$R(f) = \mathrm{E}\left\{Q(f(x))\right\}$$

What we have

- A set of samples $\{x_i, y_i\}$
- Empirical risk

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} Q(f(x_i))$$

uc3m

Empirical Risk Minimization

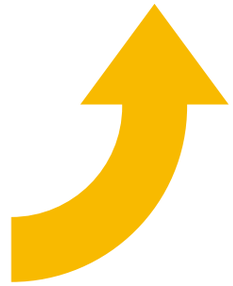
$$\min_f R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} Q(f(x_i))$$

Robust Risk Minimization

$$R(f) \leq R_{emp}(f) + \dots$$

Robust Risk Minimization

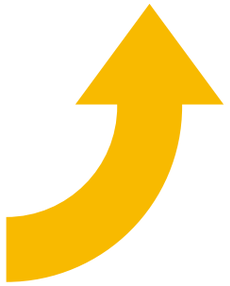
$$R(f) \leq R_{emp}(f) + \dots$$



Robust Risk Minimization

$$R(f) \leq R_{emp}(f) + \dots$$

$$\mathrm{Var}[Q(f)]$$



uc3m

Robust Risk Minimization

$$R(f) \leq R_{emp}(f) + \dots$$

$$\mathrm{Var}[Q(f)]$$

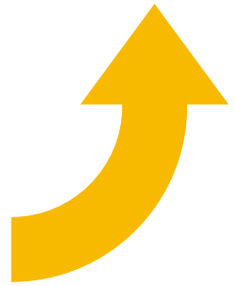
$$\min_f \left\{ \frac{1}{n} \sum_{i=1}^{n} Q(f(x_i)) \right\}$$

uc3m

Robust Risk Minimization

$$R(f) \leq R_{emp}(f) + \dots$$

$$\mathrm{Var}[Q(f)]$$



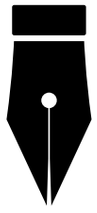
$$\min_{f} \left\{ \frac{1}{n} \sum_{i=1}^{n} Q(f(x_i)) + g(\mathrm{Var}[Q(f(x_i))]) \right\}$$

uc3m

Variance-based robust regularization

(Namkoong & Duchi, 2017) proposed...

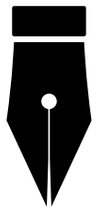
Variance-based robust regularization



(Namkoong & Duchi, 2017) proposed...

- **Risk minimization alternative**

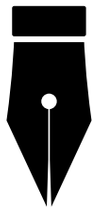
Variance-based robust regularization



(Namkoong & Duchi, 2017) proposed...

- **Risk minimization alternative**
- **Optimal and computationally efficient solution**

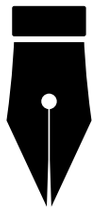
Variance-based robust regularization



(Namkoong & Duchi, 2017) proposed...

- **Risk minimization alternative**
- **Optimal and computationally efficient solution**
- **Tractable convex formulation**

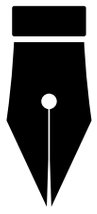
Variance-based robust regularization



(Namkoong & Duchi, 2017) proposed...

- **Risk minimization alternative**
- **Optimal and computationally efficient solution**
- **Tractable convex formulation**
- **Regularization of empirical risk by adding a variance term**

Variance-based robust regularization



(Namkoong & Duchi, 2017) proposed...

- **Risk minimization alternative**
- **Optimal and computationally efficient solution**
- **Tractable convex formulation**
- **Regularization of empirical risk by adding a variance term**

Bias ↔ Variance

uc3m

Alternative as a *min-max problem*

1. Minimization of the empirical risk $\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} p_i \cdot q_i(\theta, x_i)$

uc3m

Alternative as a *min-max problem*

1. Minimization of the empirical risk $\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} p_i \cdot q_i(\theta, x_i)$

$\max_{p} \sum_{i=1}^{n} p_i \cdot q_i$ **2. Maximization of the robust objective**

uc3m

Alternative as a *min-max problem*

1. Minimization of the empirical risk $\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} p_i \cdot q_i(\theta, x_i)$

$\max_{p} \sum_{i=1}^{n} p_i \cdot q_i$ **2. Maximization of the robust objective**

uc3m

Alternative as a *min-max problem*

Constraints

Alternative as a *min-max problem*

Constraints

$$p \in \mathcal{P}_n = \left\{ p \in \mathbb{R}^n_+ : \frac{1}{2} \left\| np - \mathbf{1} \right\|_2^2 \leq \rho, \langle \mathbf{1}, p \rangle = 1 \right\}$$

uc3m

Alternative as a *min-max problem*

Results

Alternative as a *min-max problem*

Results

- **Faster rates of convergence**

uc3m

Alternative as a *min-max problem*

Results

- Faster rates of convergence
- Improvement of out-of-sample test performance

Application on Deep Learning

Application on Deep Learning

Alternative to risk minimization

Application on Deep Learning

Alternative to risk minimization

Deep Learning method

Application on Deep Learning

Alternative to risk minimization

Deep Learning method

=

=

uc3m

Application on Deep Learning

Alternative to risk minimization

Deep Learning method

=

Cost function

Application on Deep Learning

Alternative to risk minimization

Deep Learning method

=

Cost function

+

Application on Deep Learning

Alternative to risk minimization

Deep Learning method

=

Cost function

+

Optimization procedure

Application on Deep Learning

Alternative to risk minimization

Deep Learning method

=

Cost function

+

Optimization procedure

+

Application on Deep Learning

Alternative to risk minimization

Deep Learning method

=

Cost function

+

Optimization procedure

+

Model

Convolutional Neural Network (CNN)

Image classification task

uc3m

Convolutional Neural Network (CNN)

Datasets

Image classification task

uc3m

Convolutional Neural Network (CNN)

Datasets

Image classification task

Convolutional Neural Network (CNN)

Datasets

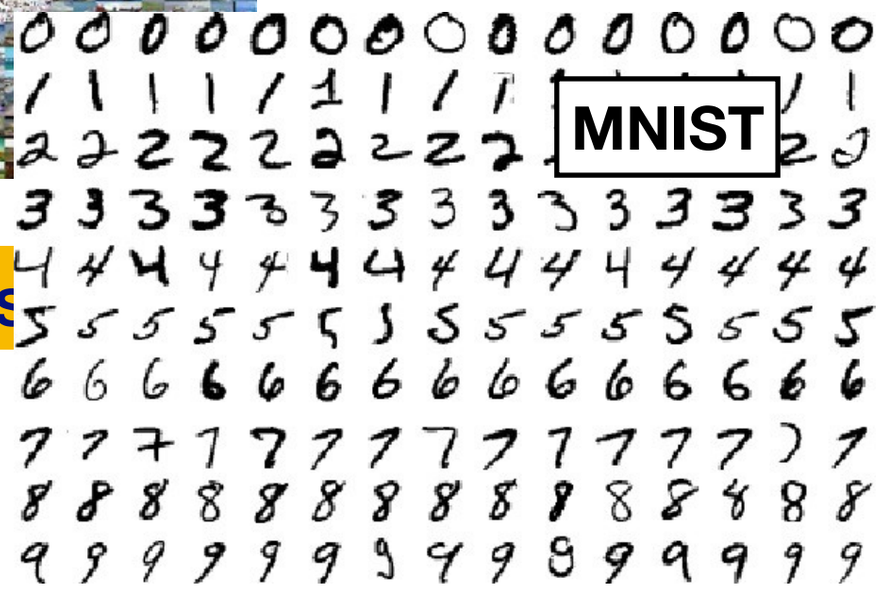
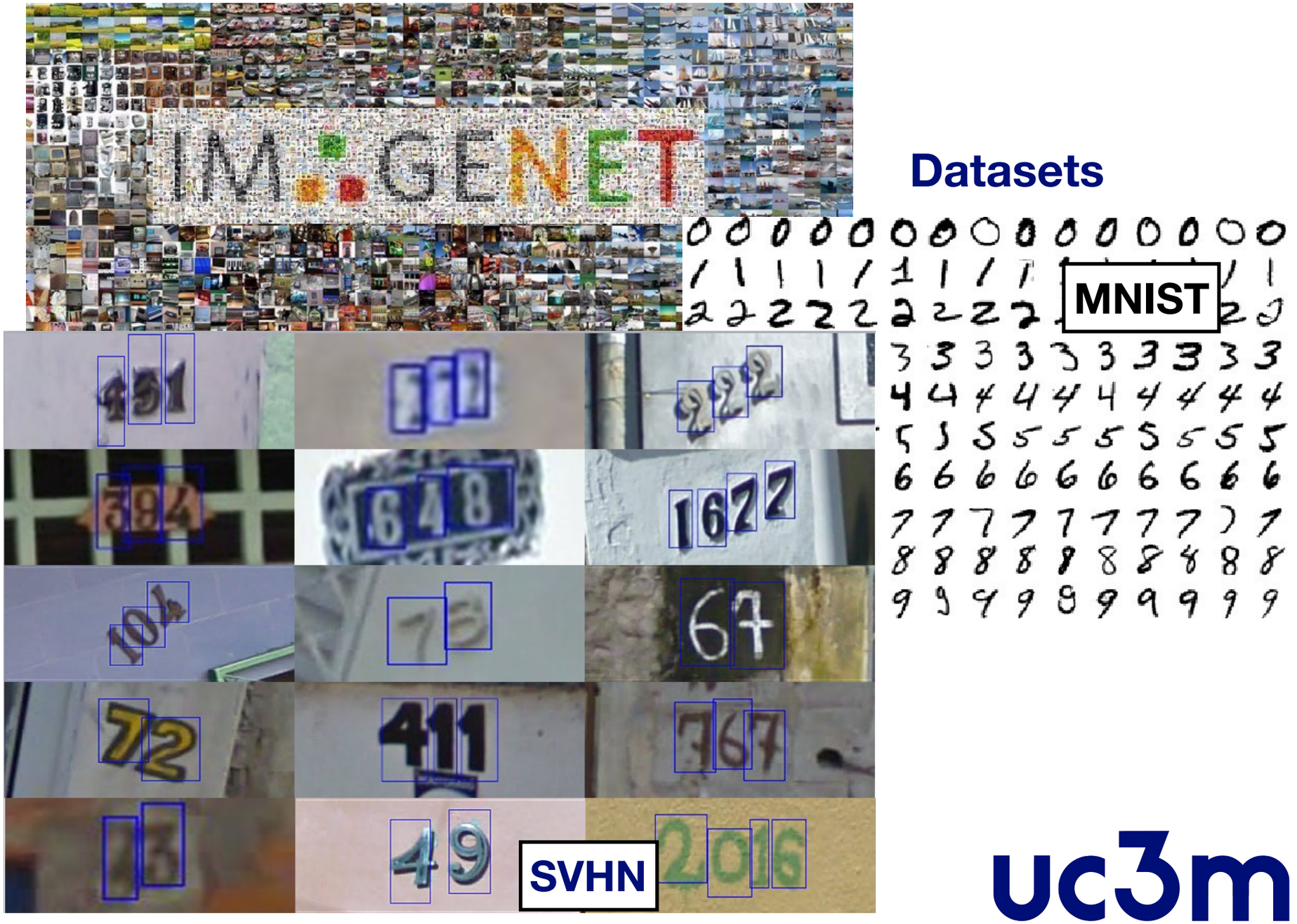


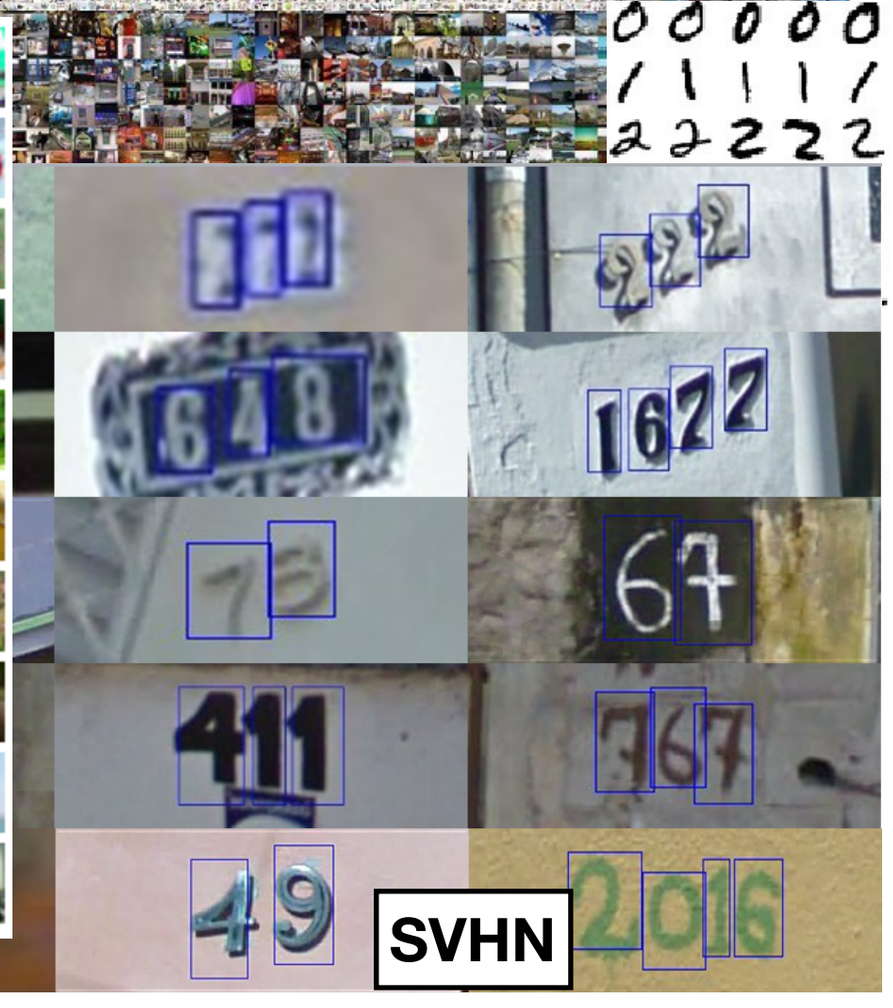
Image classification tas

Convolutional Neural Network (CNN)



Convolutional Neural Network (CNN)

Datasets

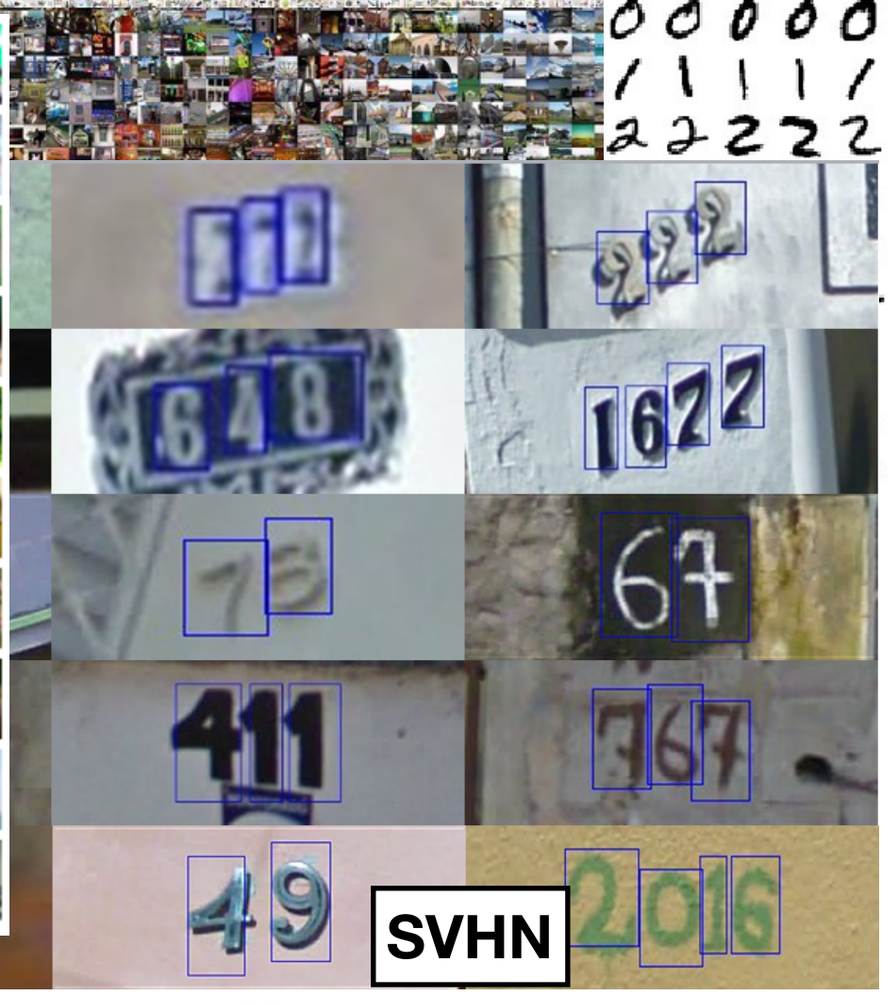


uc3m

Convolutional Neural Network (CNN)

Networks

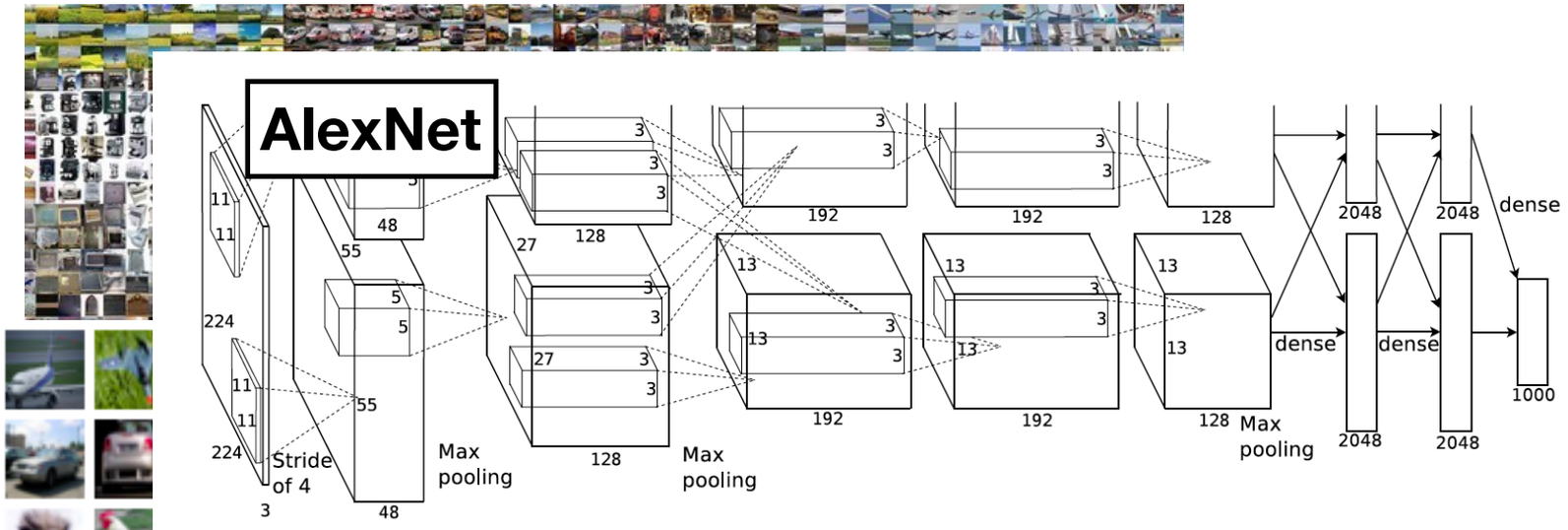
Datasets

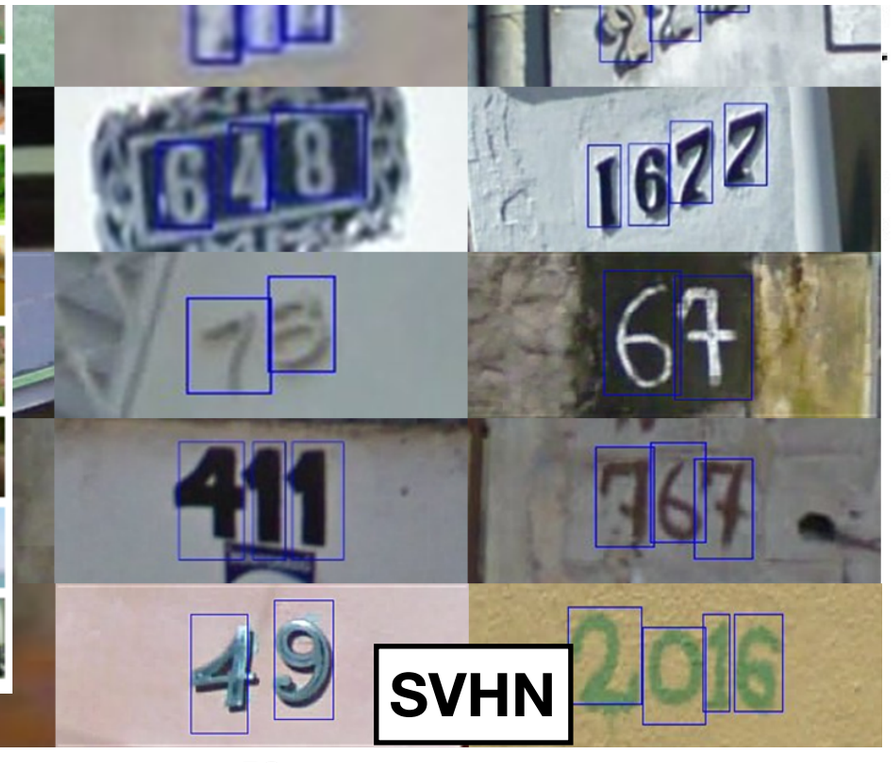


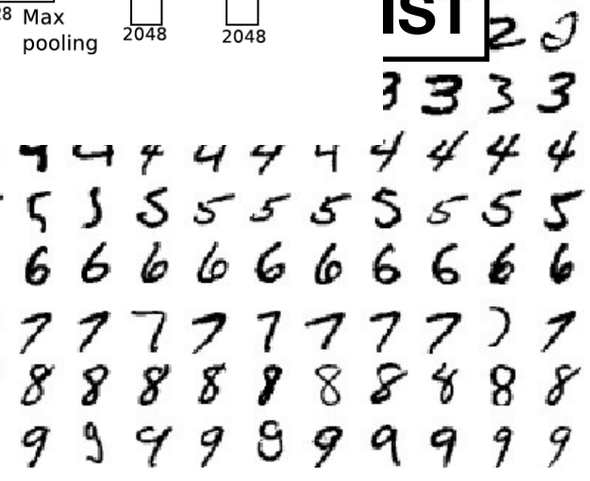
uc3m

Convolutional Neural Network (CNN)

Networks



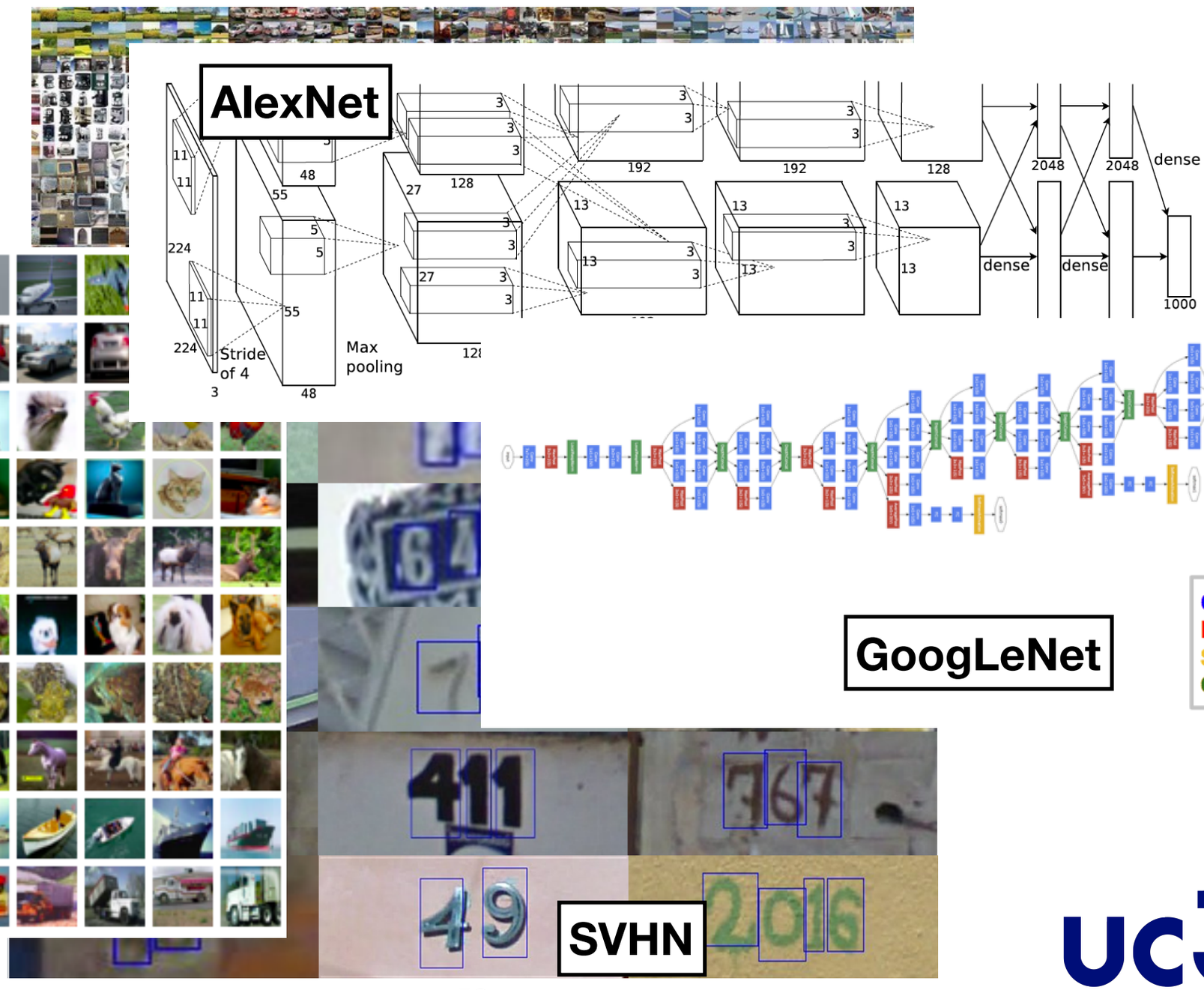




uc3m

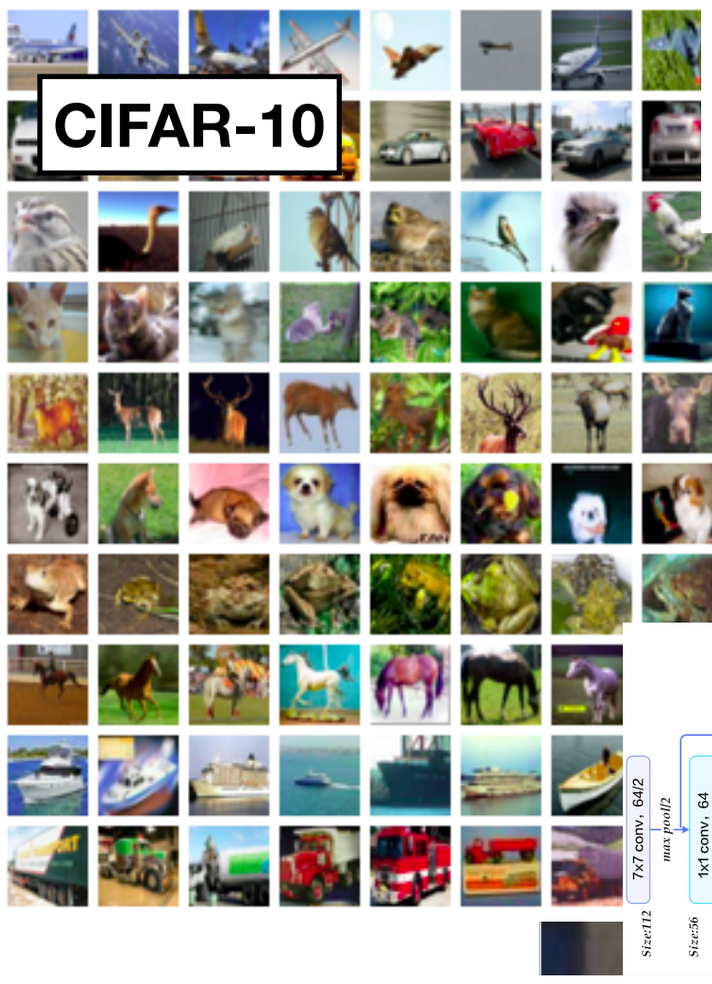
Convolutional Neural Network (CNN)

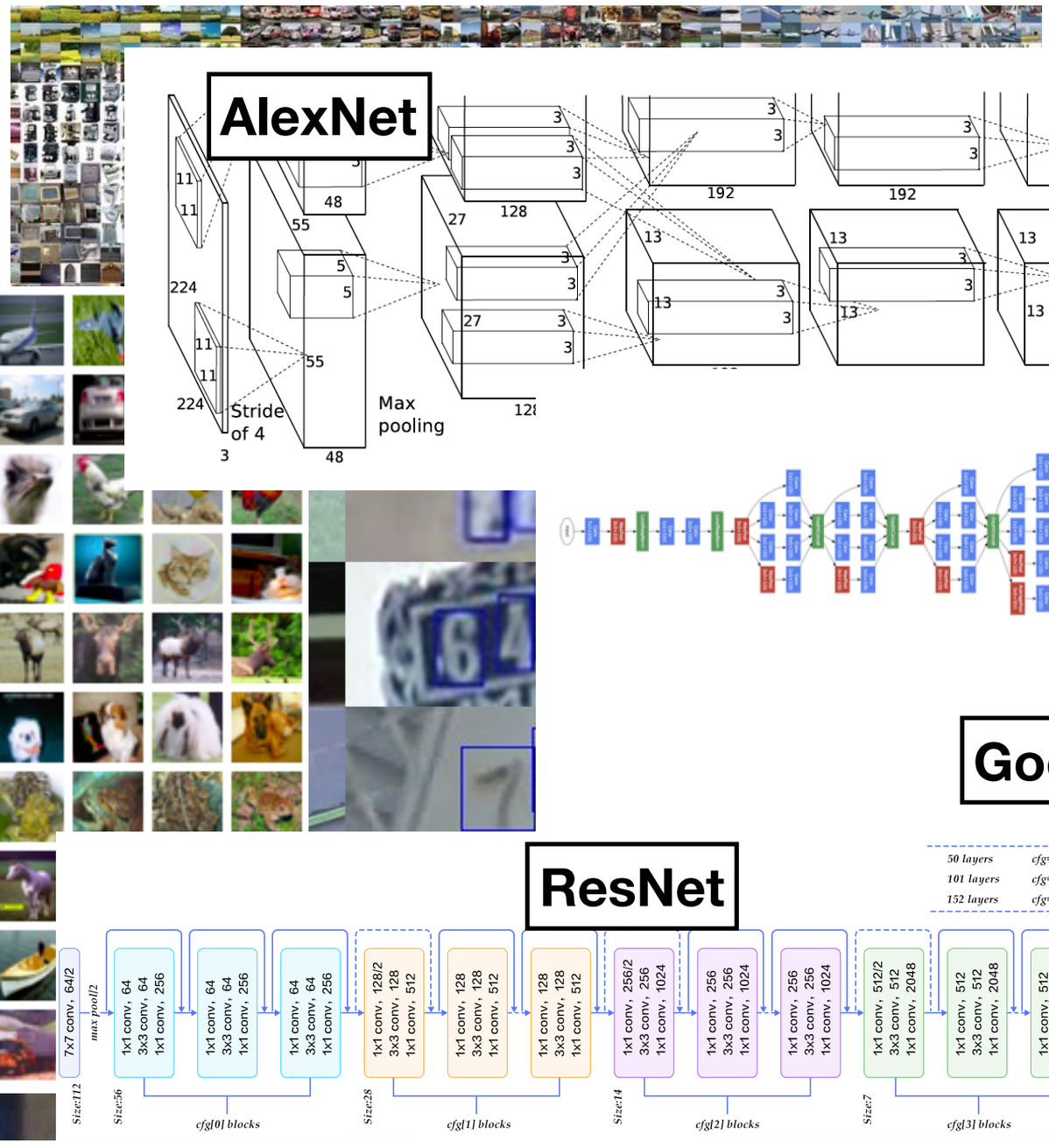
Networks

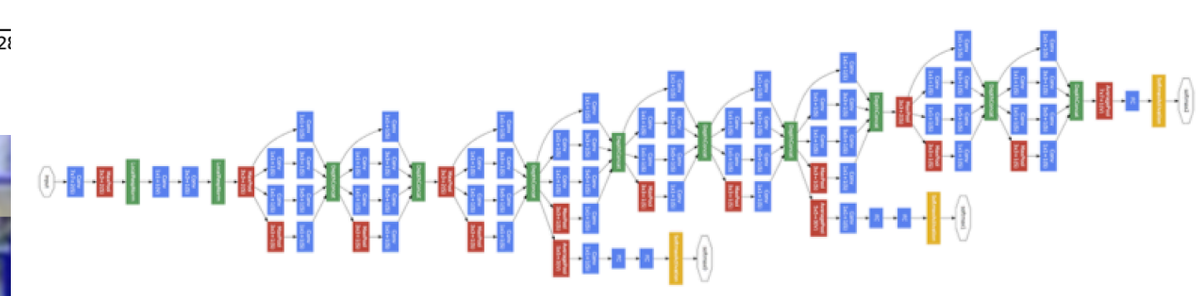


Convolutional Neural Network (CNN)

Networks



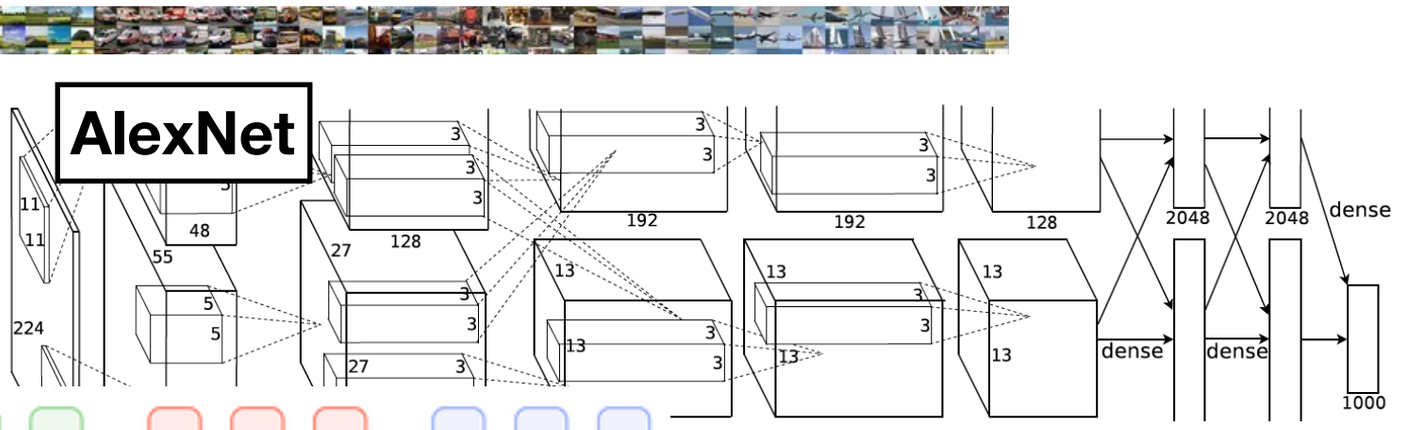


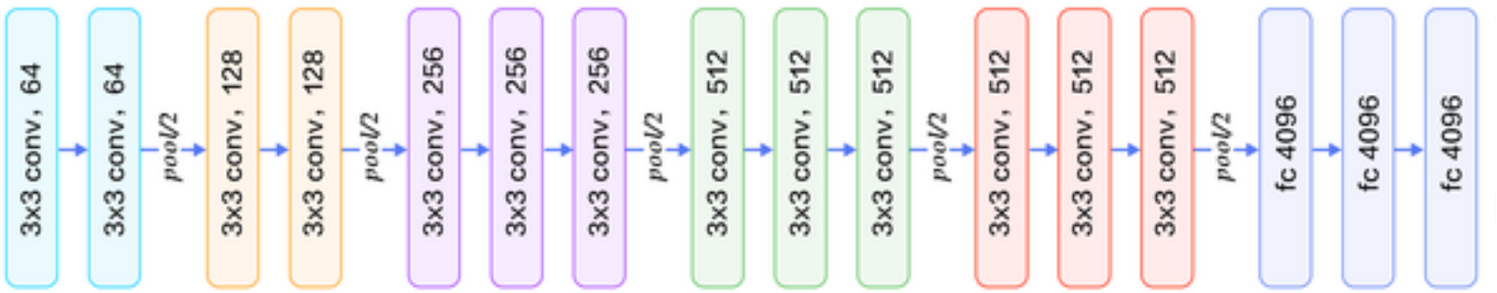


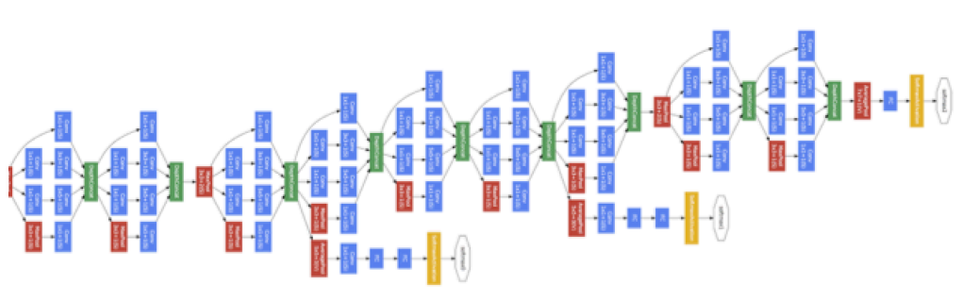
Convolution
Pooling
Softmax
Other

Convolutional Neural Network (CNN)

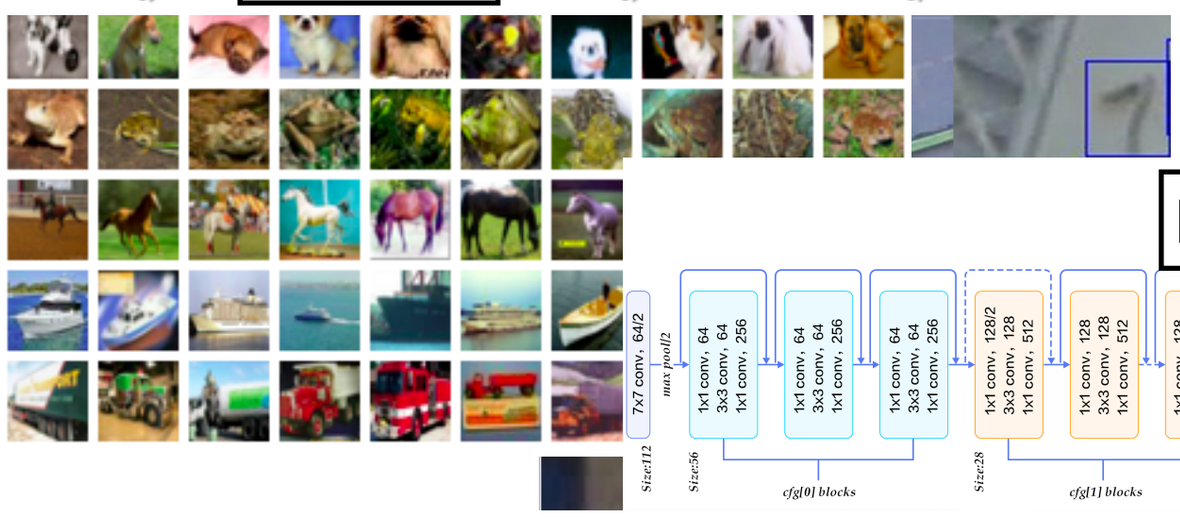
Networks







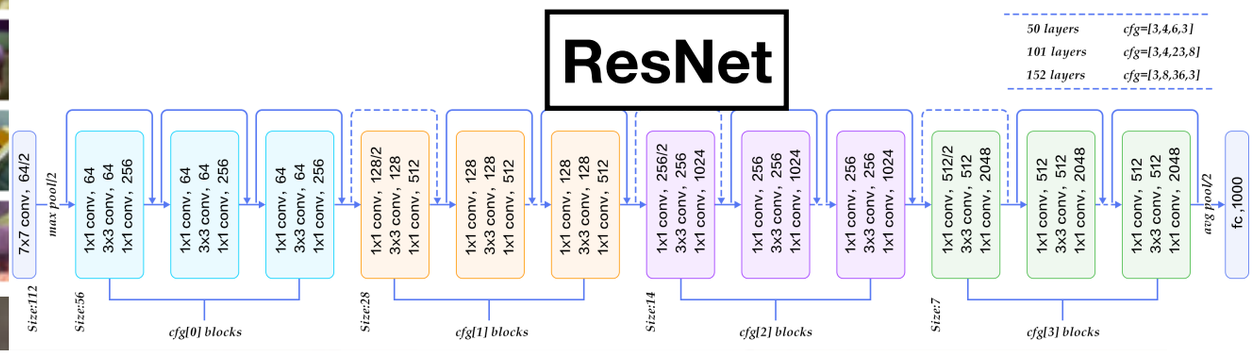
VGGNet



GoogLeNet

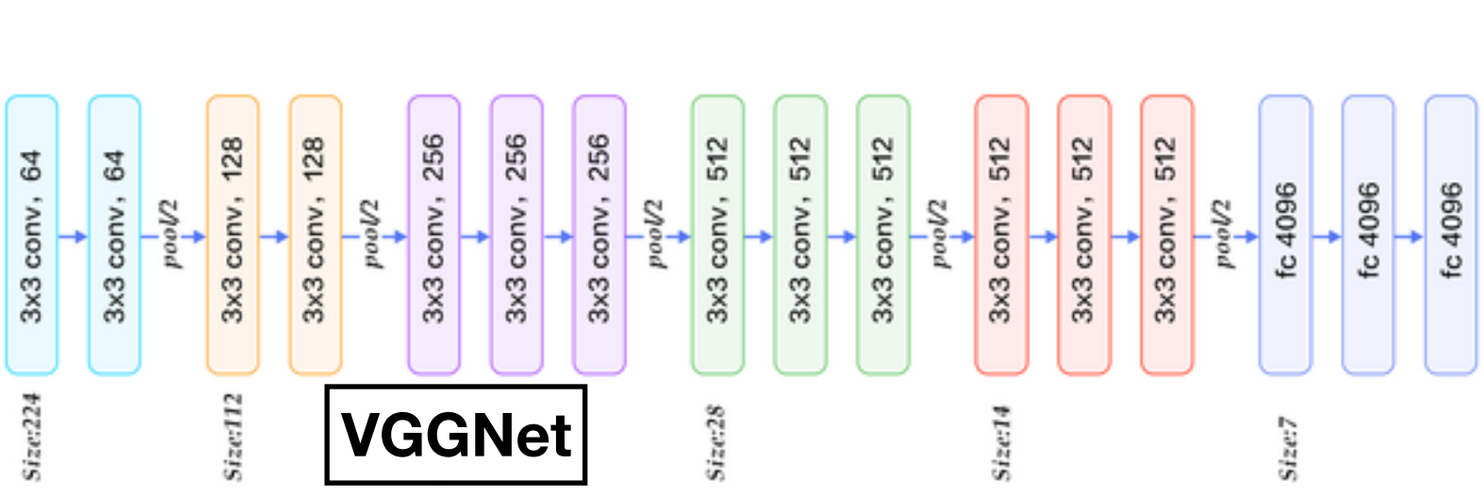
Convolution
Pooling
Softmax
Other

ResNet



Convolutional Neural Network (CNN)

Networks



Datasets

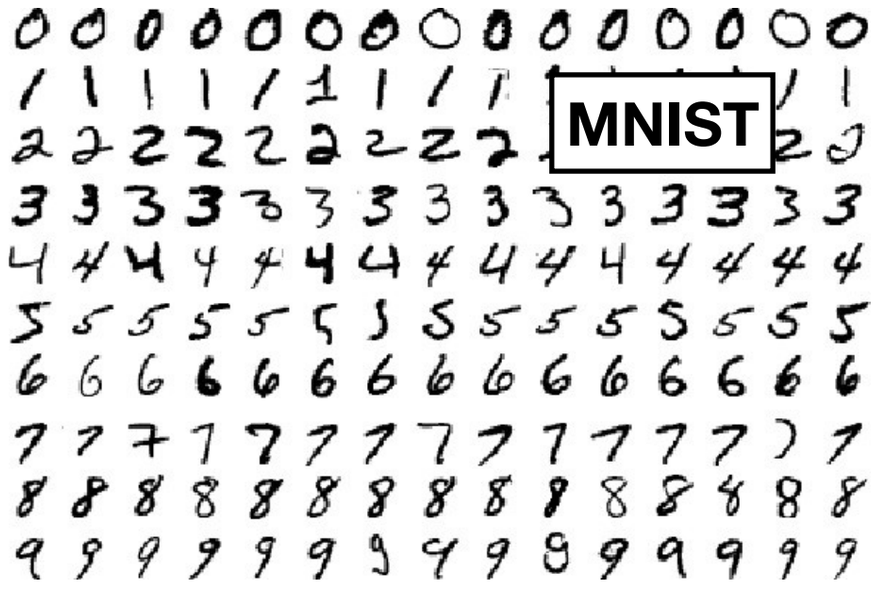


Image classification task

Application on Deep Learning

Alternative to risk minimization

Deep Learning method

=

Cost function

+

Optimization procedure

+

Model

Application on Deep Learning

Alternative to risk minimization

Deep Learning method

=

Cost function

+

Optimization procedure

+

Model

Application on Deep Learning

Alternative to risk minimization

10^6 - 10^9 parameters
10^5 - 10^6 samples

Deep Learning method

Cost function

+

Optimization procedure

+

Model

Application on Deep Learning

Alternative to risk minimization

Computational cost

Deep Learning method

Cost function

+

Optimization procedure

+

Model

Variance reducer regularization

Applied to Mini-batch…

Variance reducer regularization

Applied to Mini-batch…

Traditional methods

Variance reducer regularization

Applied to Mini-batch…

Traditional methods

- **Sample uniformly**

uc3m

Variance reducer regularization

Applied to Mini-batch…

Traditional methods

- **Sample uniformly**
- **Unbiased estimation**

uc3m

Variance reducer regularization

Applied to Mini-batch…

Traditional methods

- Sample uniformly
- Unbiased estimation
- High variance

uc3m

Variance reducer regularization

Applied to Mini-batch…

Traditional methods

- Sample uniformly
- Unbiased estimation
- High variance

Robust solution

Variance reducer regularization

Applied to Mini-batch…

Traditional methods

- Sample uniformly
- Unbiased estimation
- High variance

Robust solution

- Non-uniform sampling

uc3m

Variance reducer regularization

Applied to Mini-batch…

Traditional methods

- Sample uniformly
- Unbiased estimation
- High variance

Robust solution

- Non-uniform sampling
- Balance bias-variance

Variance reducer regularization

Applied to Mini-batch…

Traditional methods

- Sample uniformly
- Unbiased estimation
- High variance

Robust solution

- Non-uniform sampling
- Balance bias-variance
- Reduce variance

Methodology

Applied to Mini-batch…

Methodology

Applied to Mini-batch…

1. **VR-M: Repeat a percentage of the worst performed samples from one mini-batch to the next one.**

Methodology

Applied to Mini-batch…

1. **VR-M: Repeat a percentage of the worst performed samples from one mini-batch to the next one.**

2. **VR-E: Repeat a percentage of the worst performed samples from one epoch to the next one.**

uc3m

VR-M vs VR-E

Algorithm 1 Variance Reducer per Mini-batch (VR-M)

Input: Datasets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$.
Output: Test accuracy $\boldsymbol{\eta}$.

Initialize parameters $\boldsymbol{\theta}$, number of epochs E and repetition rate ϵ;
for $e = 1 \dots E$ **do**
  Divide dataset $\mathcal{D}_{\text{train}}$ in M mini-batches;
  for $m = 1 \dots M$ **do**
    $\{\mathbf{x}_m, \mathbf{y}_m\} \leftarrow$ Obtain next mini-batch m;
    $\{\mathbf{x}_m, \mathbf{y}_m\} \leftarrow$ Substitute $\epsilon \cdot M$ samples with the $\{\mathbf{x}_{m-1}, \mathbf{y}_{m-1}\}$ of highest $\boldsymbol{\ell}_{m-1}$;
    $\boldsymbol{\ell}_m \leftarrow$ Evaluate cross-entropy in mini-batch m;
    $\boldsymbol{\theta} \leftarrow$ Update parameters with stochastic gradient descent (SGD);
  end for
  $\eta_e \leftarrow$ Compute test accuracy on $\mathcal{D}_{\text{test}}$;
  Shuffle $\mathcal{D}_{\text{train}}$;
end for

Algorithm 2 Variance Reducer per Epoch (VR-E)

Input: Datasets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$.
Output: Test accuracy $\boldsymbol{\eta}$.

Initialize parameters $\boldsymbol{\theta}$, number of epochs E and repetition rate ϵ;
Initialize $\mathcal{D}^{(1)}_{\text{train}} = \mathcal{D}_{\text{train}}$;
for $e = 1 \dots E$ **do**
  Divide dataset $\mathcal{D}^{(e)}_{\text{train}}$ in M mini-batches;
  for $m = 1 \dots M$ **do**
    $\{\mathbf{x}_m, \mathbf{y}_m\} \leftarrow$ Obtain next mini-batch m;
    $\boldsymbol{\ell}^{(e)}_m \leftarrow$ Evaluate cross-entropy in mini-batch m;
    $\boldsymbol{\theta} \leftarrow$ Update parameters with stochastic gradient descent (SGD);
  end for
  $\eta_e \leftarrow$ Compute test accuracy on $\mathcal{D}_{\text{test}}$;
  Shuffle $\mathcal{D}_{\text{train}}$;
  $\mathcal{D}^{(e+1)}_{\text{train}} \leftarrow \mathcal{D}_{\text{train}}$;
  $\mathcal{D}^{(e+1)}_{\text{train}} \leftarrow$ Substitute $\epsilon \cdot E$ samples with $\{x_i, y_i\} \in \mathcal{D}^{(e)}_{\text{train}}$ of highest $\boldsymbol{\ell}^{(e)}$;
end for

uc3m

Methodology

Applied to Mini-batch…

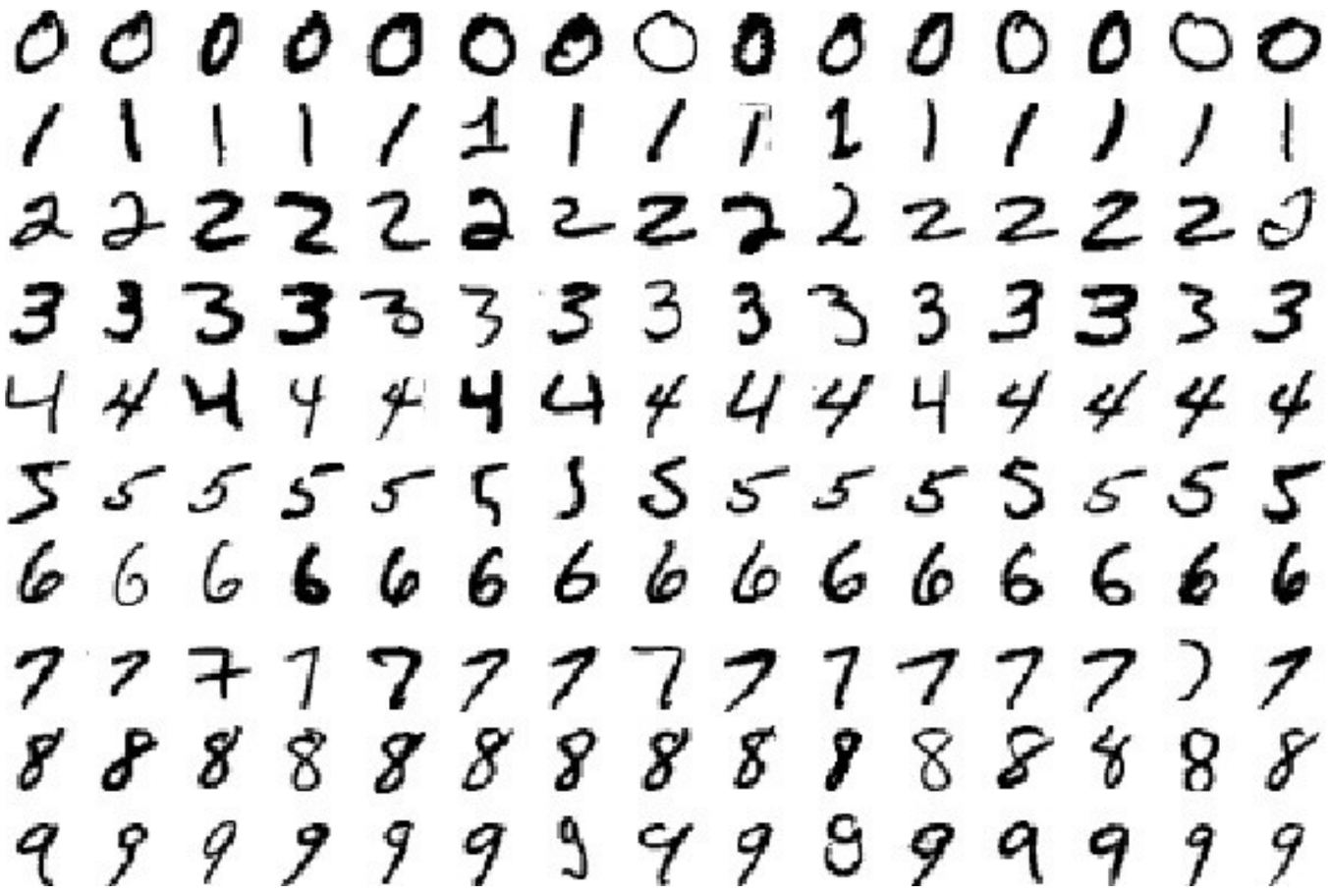
1. **VR-M: Repeat a percentage of the worst performed samples from one mini-batch to the next one.**

2. **VR-E: Repeat a percentage of the worst performed samples from one epoch to the next one.**

3. **PVR-M/E: Resample half of the data points from the worst performed ones each mini-batch or epoch.**

uc3m

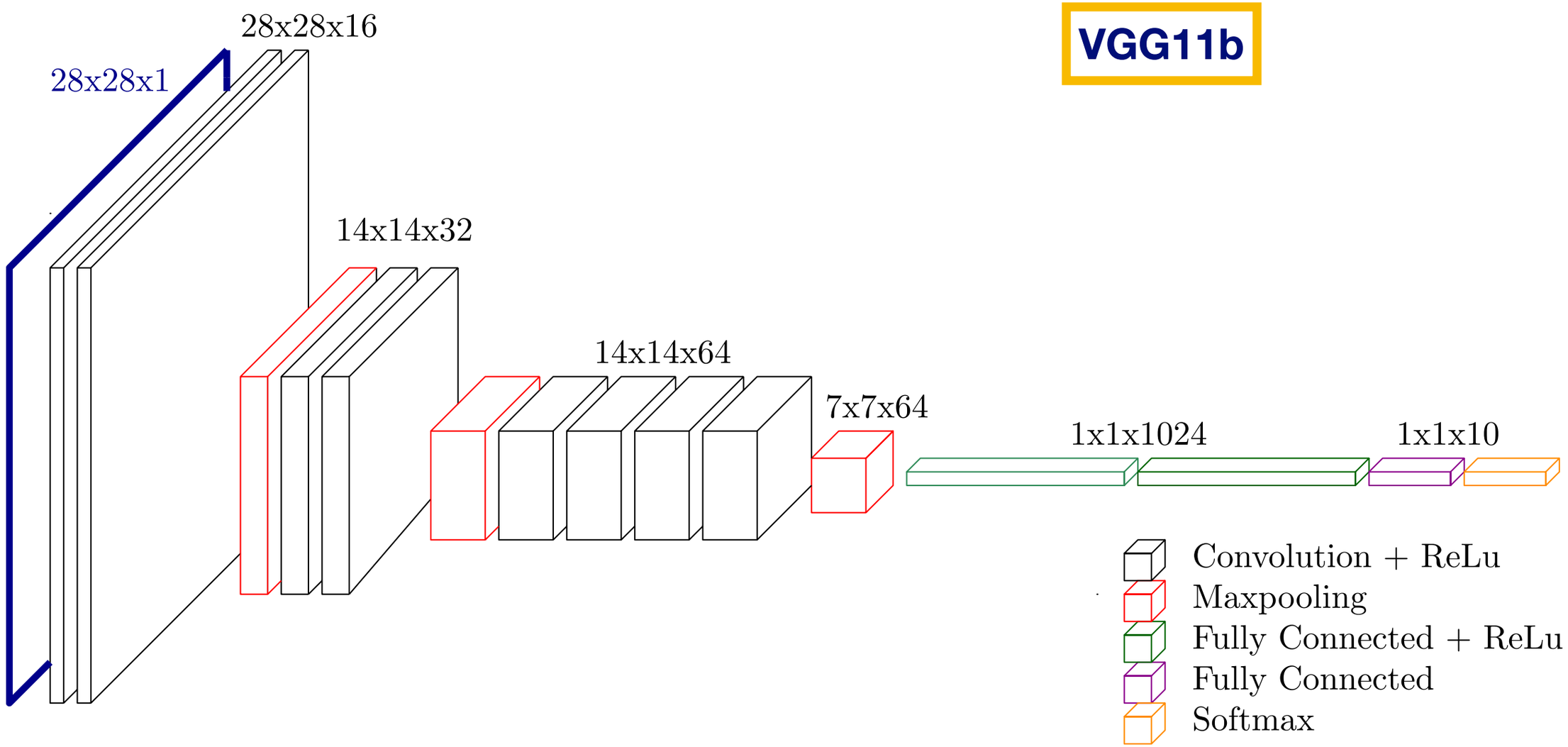
MNIST Dataset



- ➡ 60000 training images
- ➡ 10000 test images
- ➡ 28x28 pixels
- ➡ Gray scale

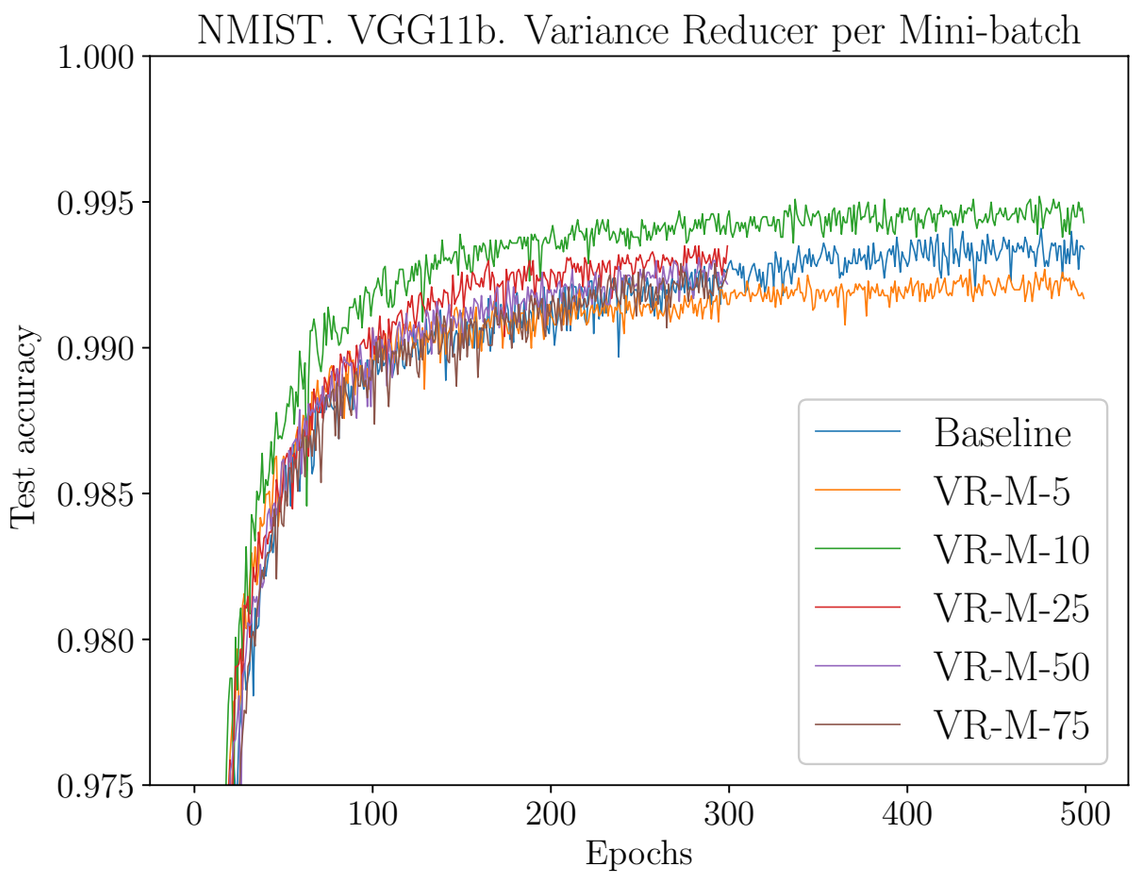
uc3m

Very deep CNN

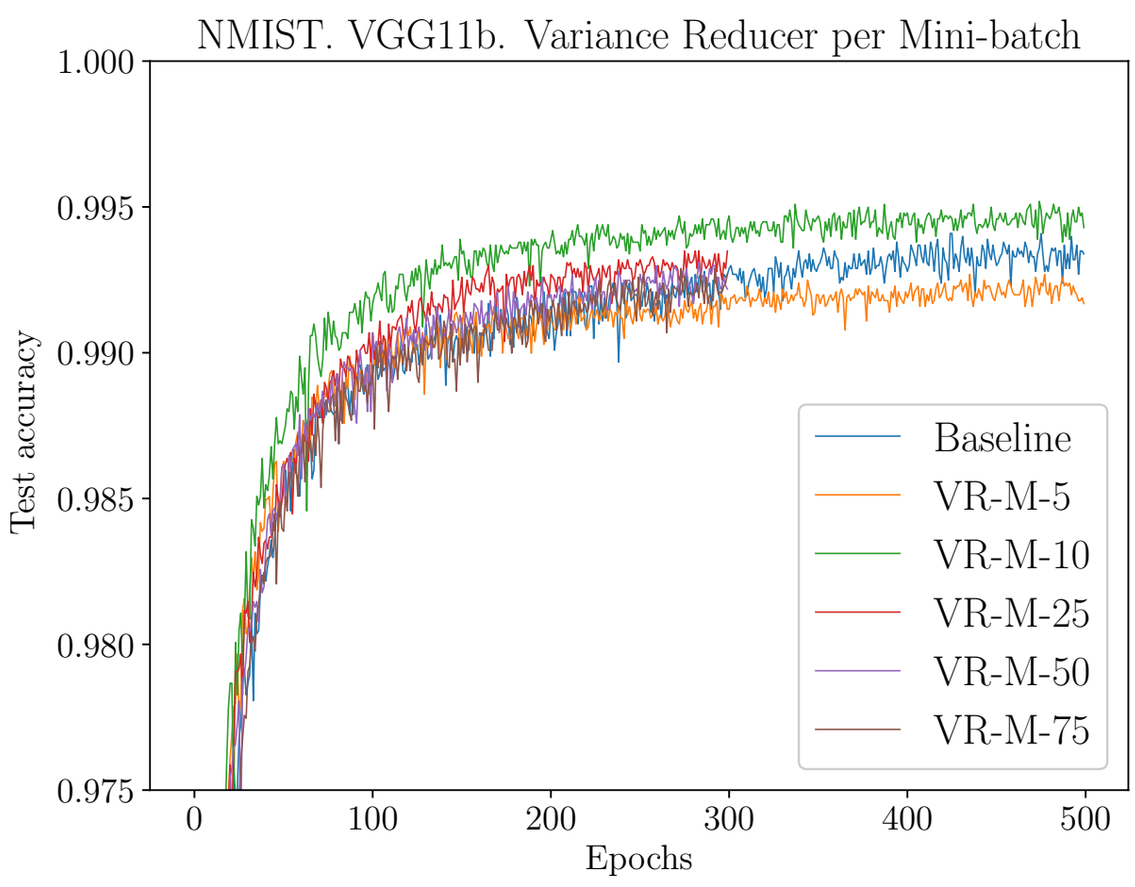


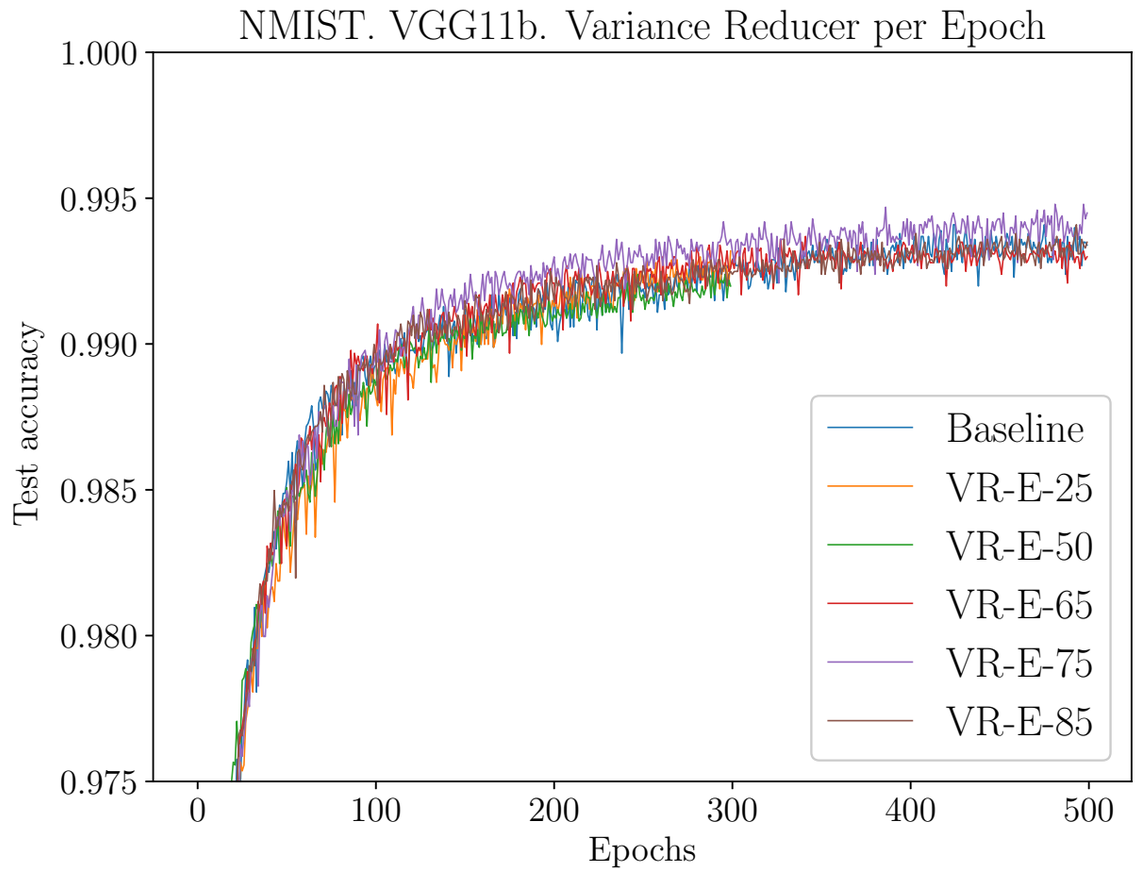
Based on (Simonyan & Zisserman, 2015)

Results

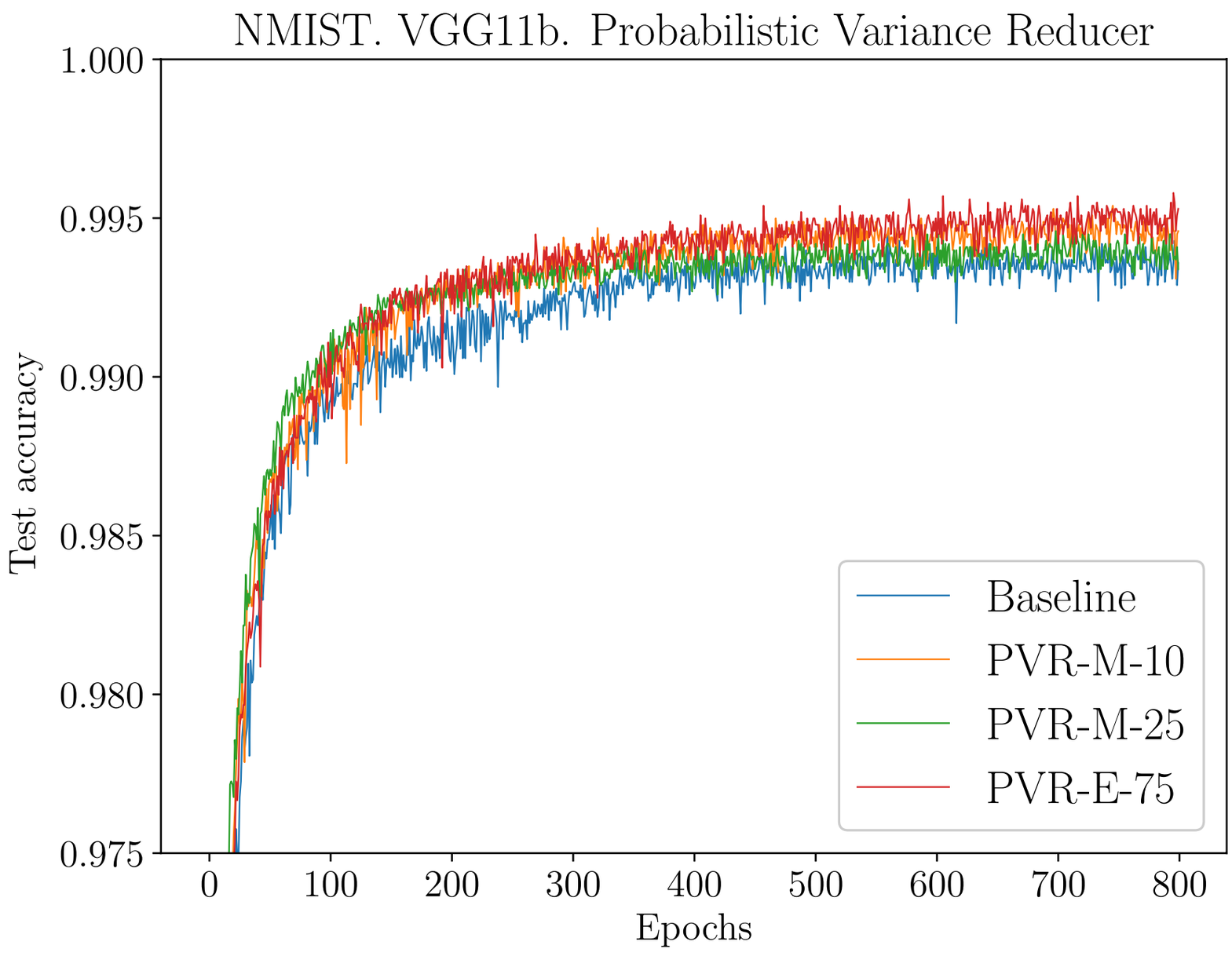

NMIST. VGG11b. Variance Reducer per Mini-batch
Test accuracy
Epochs
1.000
0.995
0.990
0.985
0.980
0.975
0
100
200
300
400
500
Baseline
VR-M-5
VR-M-10
VR-M-25
VR-M-50
VR-M-75

Results

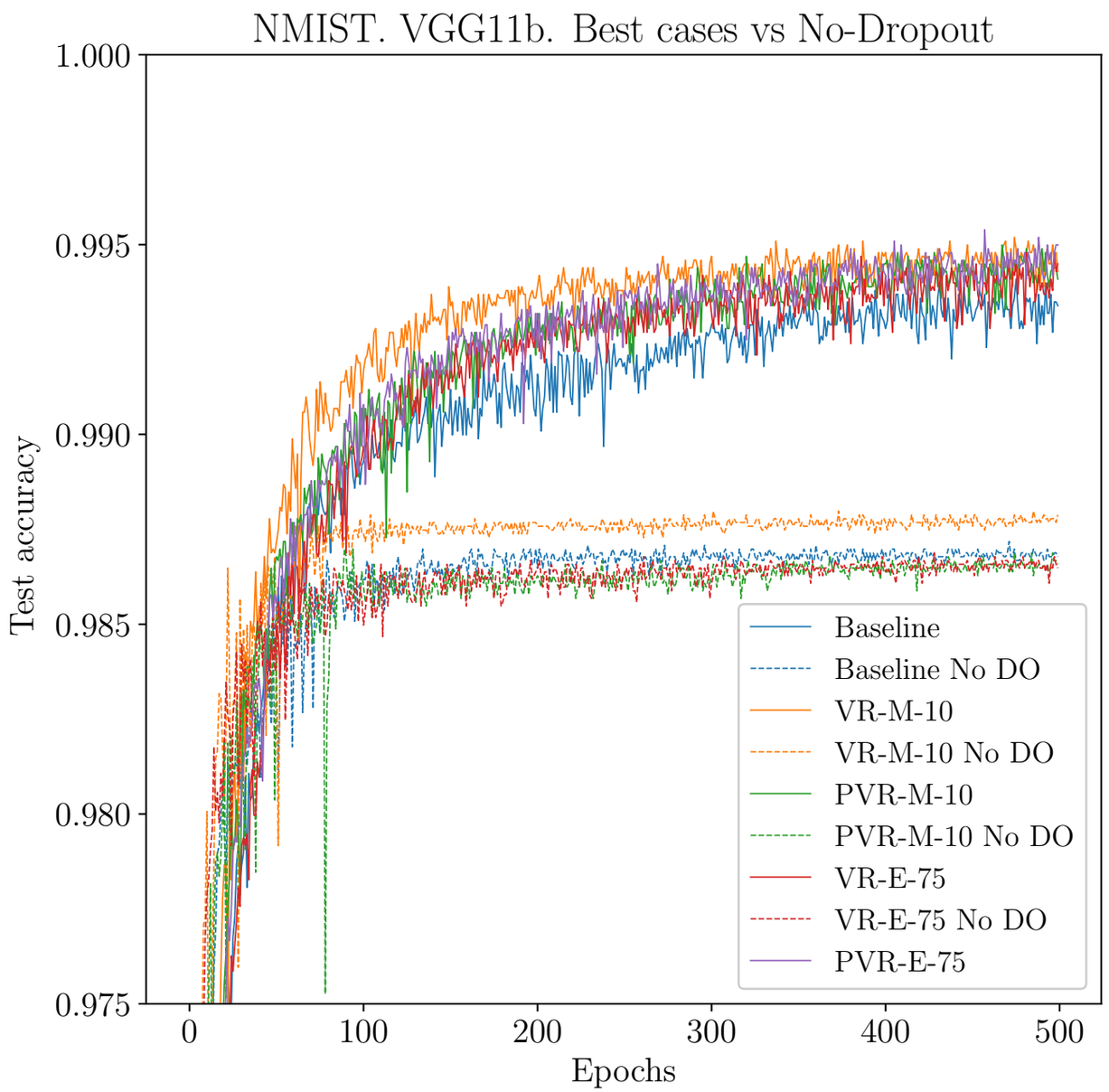

NMIST. VGG11b. Variance Reducer per Mini-batch
Test accuracy
Epochs
1.000
0.995
0.990
0.985
0.980
0.975
0
100
200
300
400
500
Baseline
VR-M-5
VR-M-10
VR-M-25
VR-M-50
VR-M-75


NMIST. VGG11b. Variance Reducer per Epoch
Test accuracy
Epochs
1.000
0.995
0.990
0.985
0.980
0.975
0
100
200
300
400
500
Baseline
VR-E-25
VR-E-50
VR-E-65
VR-E-75
VR-E-85

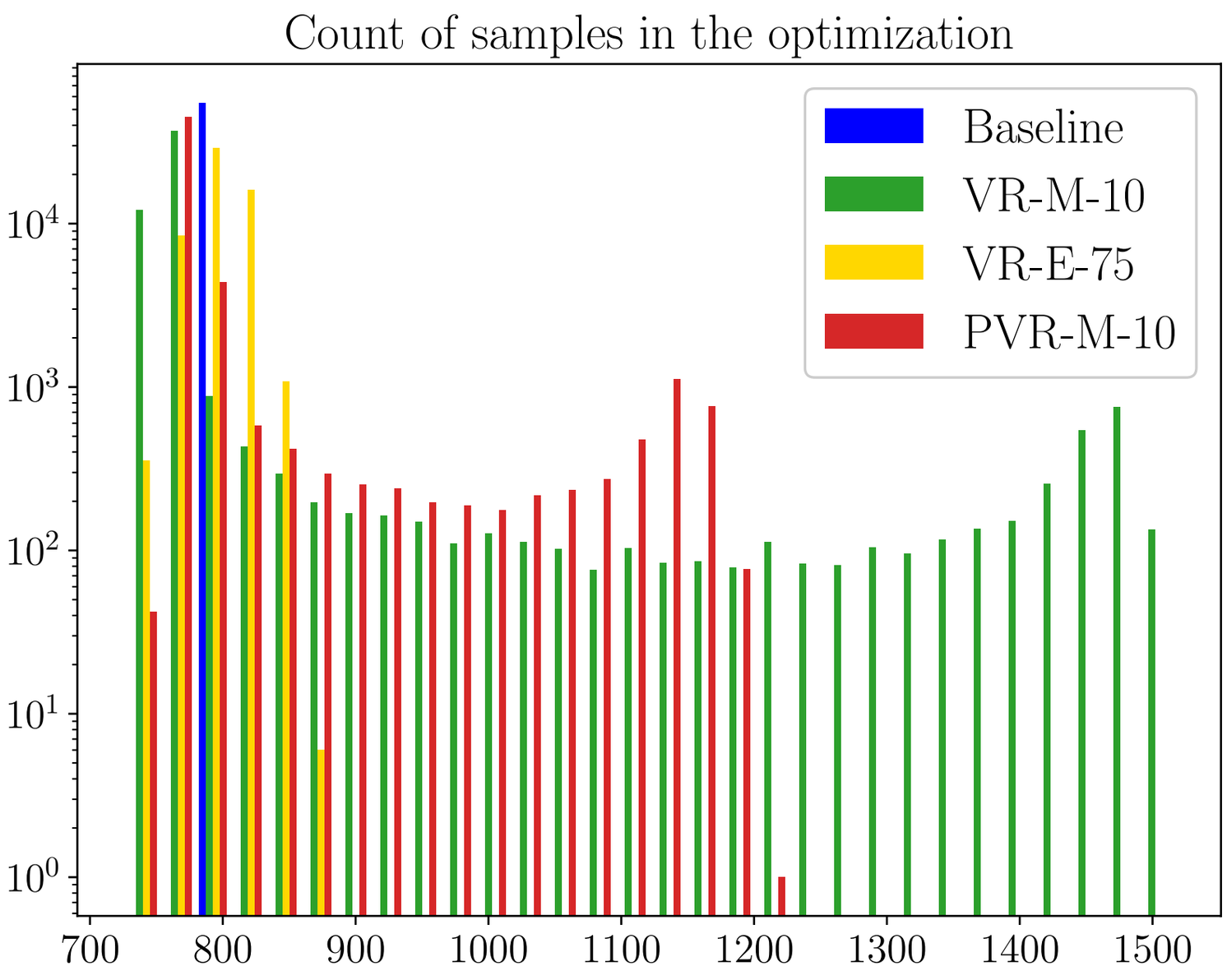
Results


NMIST. VGG11b. Probabilistic Variance Reducer
Test accuracy
1.000
0.995
0.990
0.985
0.980
0.975
0
100
200
300
400
500
600
700
800
Epochs
Baseline
PVR-M-10
PVR-M-25
PVR-E-75

Results


NMIST. VGG11b. Best cases vs No-Dropout
Test accuracy
Epochs
1.000
0.995
0.990
0.985
0.980
0.975
0
100
200
300
400
500
Baseline
Baseline No DO
VR-M-10
VR-M-10 No DO
PVR-M-10
PVR-M-10 No DO
VR-E-75
VR-E-75 No DO
PVR-E-75

Results


Count of samples in the optimization
Baseline
VR-M-10
VR-E-75
PVR-M-10
10^4
10^3
10^2
10^1
10^0
700
800
900
1000
1100
1200
1300
1400
1500

Conclusions

✤ **Study of a novel robust estimation based on a reduction of the variance.**

Conclusions

✤ Study of a novel robust estimation based on a reduction of the variance.

✤ Prove theoretical evidences with experiments in CNN.

Conclusions

- ✤ Study of a novel robust estimation based on a reduction of the variance.
- ✤ Prove theoretical evidences with experiments in CNN.
- ✤ Obtain methods with different characteristics:
  - VR-M with faster convergence.
  - VR-E with better performance.
  - PVR to deal with misclassified samples.

Conclusions

Epoch	10	100	300	500
Baseline	95,49%	98,91%	99,26%	99,33%
VR-M-10	**96,03**%	**99,10**%	**99,44**%	99,43%
PVR-M-10	95,10%	99,06%	99,37%	99,41%
VR-E-75	95,56%	98,93%	99,35%	99,45%
PVR-E-75	95,43%	98,96%	99,36%	**99,50**%

Conclusions

- Study of a novel robust estimation based on a reduction of the variance.
- Prove theoretical evidences with experiments in CNN.
- Obtain methods with different characteristics:
  - VR-M with faster convergence.
  - VR-E with better performance.
  - PVR to deal with misclassified samples.
- Open work with the study of other scenarios and larger datasets.

References

✓ H. Namkoong and J. C. Duchi. **Variance-based regularization with convex objectives**. NIPS 2017

✓ Z. Borsos, A. Krause and K. Levy. **Online variance reduction for stochastic optimization**. arXiv: 1802.04715, 2018

✓ K. Simonyan and A. Zisserman. **Very deep convolutional networks for large-scale image recognition**. In *In Proc. ICLR*, 2015.

Thank you!

uc3m