MCV MASTER IN COMPUTER VISION Barcelona

IMAGE CLASSIFICATION

Task 4: Convolutional Networks from scratch

Group 5

Marc Artero · Diego Hernández · Bernat Medina · Pau Monserrat

Module C3: Machine Learning for Computer Vision

Contents

01 Introduction

02 CNNs from scratch

03 Stage 1: Best model

04 Stage 2: Efficiency

05 Results

06 Conclusions

Contents

01 Introduction

02 CNNs from scratch

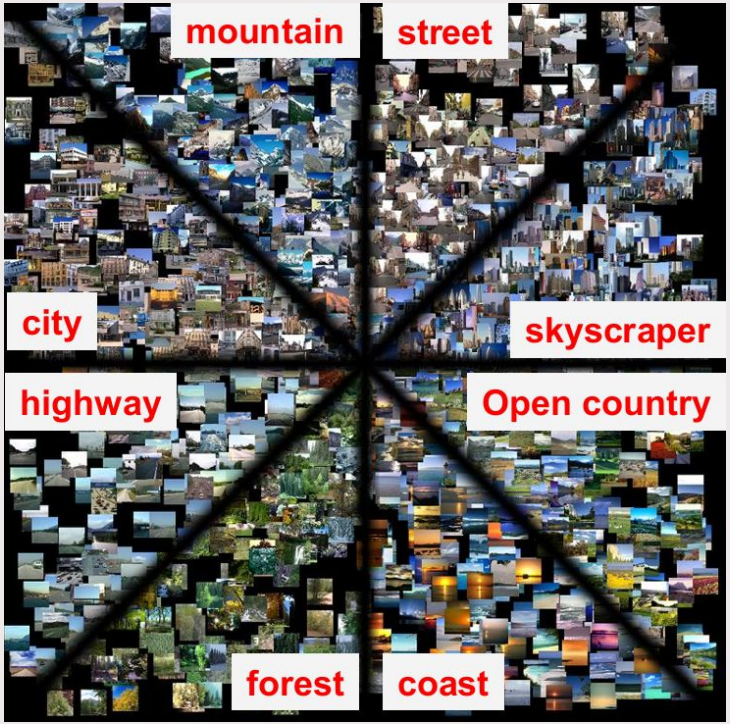
03 Stage 1: Best model

04 Stage 2: Efficiency

05 Results

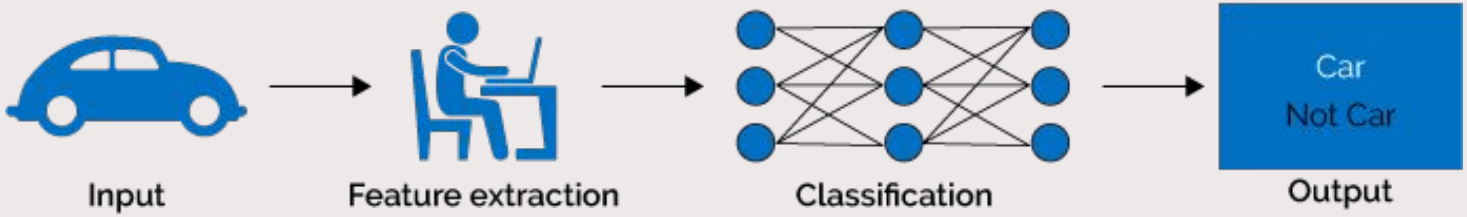
06 Conclusions

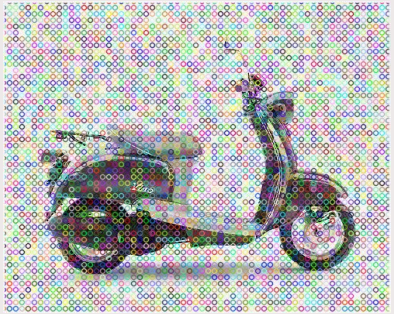
What do we aim for?

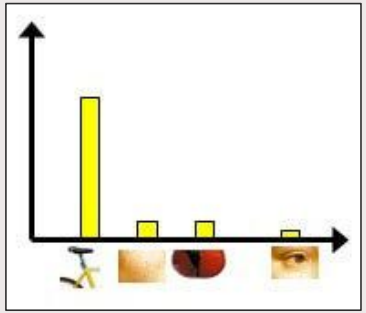


Week 1: BoVW

Handcrafted features require intensive manual engineering.



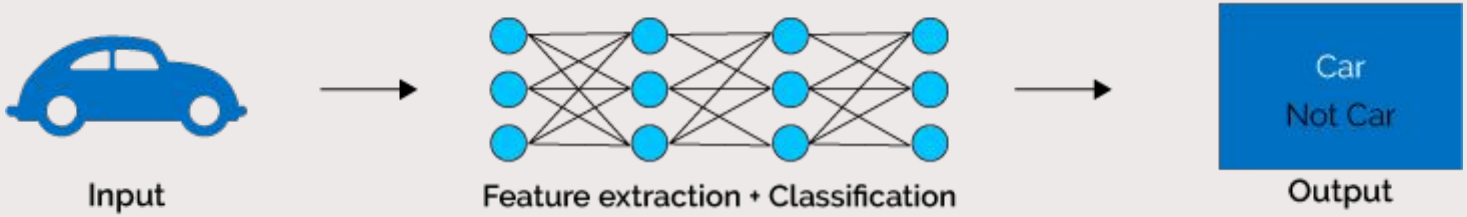


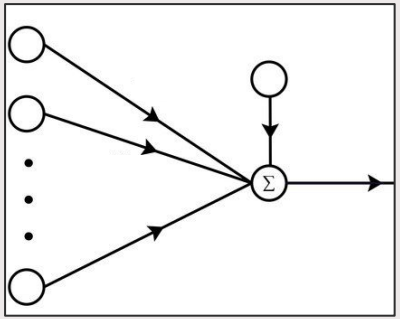


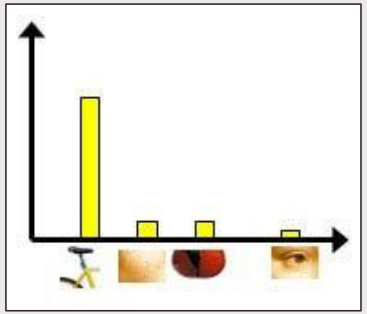
	Accuracy (%)
Week 1	88.7

Week 2: DL / BoVW

Learned features automate the pipeline, but with MLPs alone we lack spatial awareness.



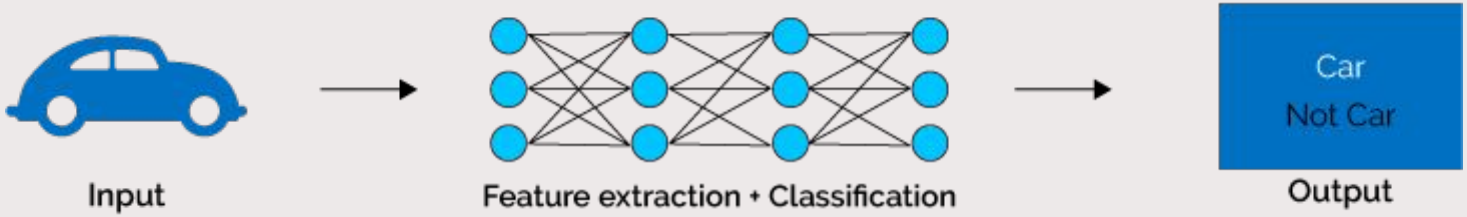


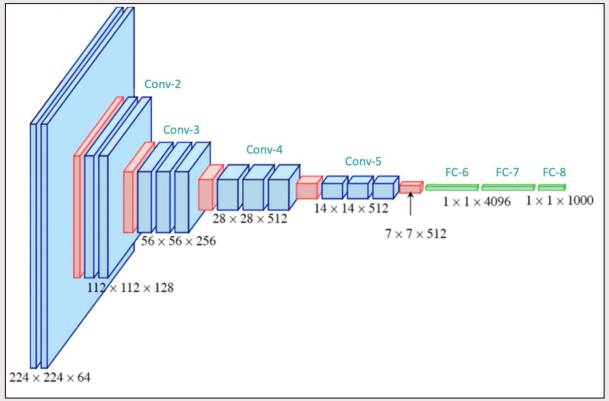


	Accuracy (%)
Week 1	88.7
Week 2	69.3

Week 3: CNNs

Fine-tuned deep CNNs solve the problem, but at the cost of massive complexity.





	Accuracy (%)
Week 1	88.7
Week 2	69.3
Week 3	92.4

Contents

01 Introduction

02 CNNs from scratch

03 Stage 1: Best model

04 Stage 2: Efficiency

05 Results

06 Conclusions

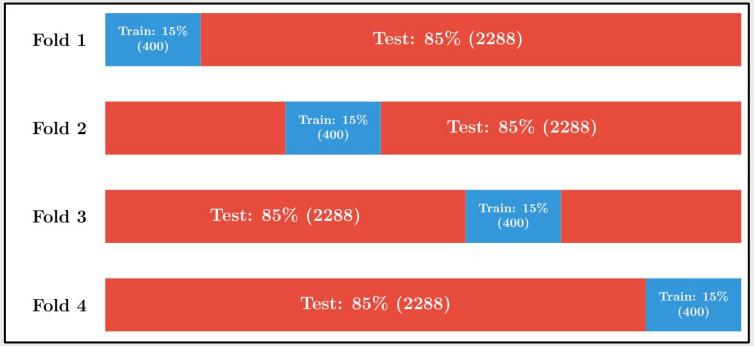
How are we tackling the problem?

Objective: Design and train a custom CNN.

Main challenges:

- Data scarcity.
- Learning features from scratch.
- Model's complexity management.

Methodology: Best model w.r.t. efficiency.



Data partitioning of the four *small* training folds.

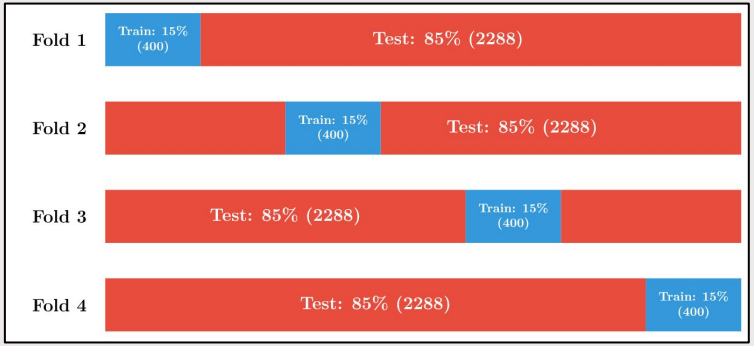
How are we tackling the problem?

Objective: Design and train a custom CNN.

Main challenges:

- Data scarcity.
- Learning features from scratch.
- Model's complexity management.

Methodology: Best model w.r.t. efficiency.



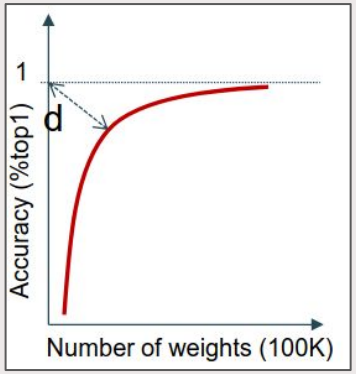
Data partitioning of the four *small* training folds.

Metrics

- Efficiency:

$$\frac{\text{Accuracy}}{\frac{\text{Number of Parameters}}{100{,}000}}$$

- Distance *d* to *(0,1)*:



Initial setup

- **Epochs**: 20.
- **Batch size**: 16.
- **Learning rate**: 0.001.
- **Optimizer**: Adam.
- **Image size**: 224x224.
- **Loss**: Cross-Entropy.
- **Strategy:** Cross-validation.

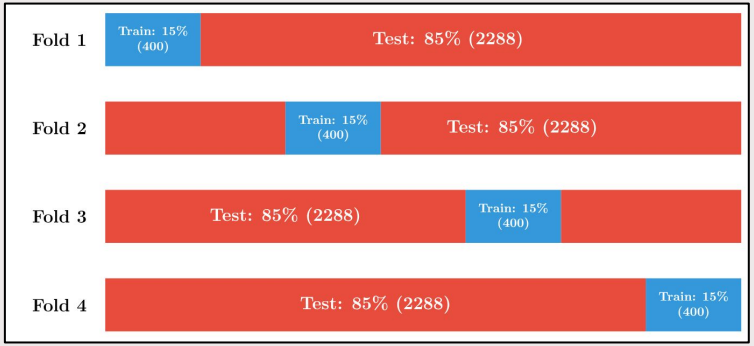
How are we tackling the problem?

Objective: Design and train a custom CNN.

Main challenges:

- Data scarcity.
- Learning features from scratch.
- Model's complexity management.

Methodology: Best model w.r.t. efficiency.



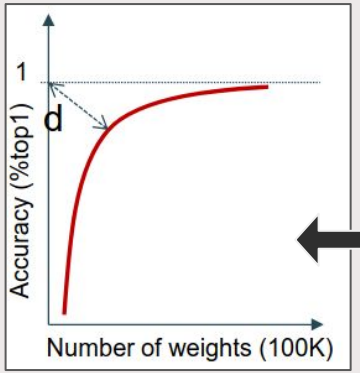
Data partitioning of the four *small* training folds.

Metrics

- Efficiency:

$$\frac{\text{Accuracy}}{\frac{\text{Number of Parameters}}{100{,}000}}$$

- Distance *d* to *(0,1)*:



We focus on this one!

Initial setup

- **Epochs**: 20.
- **Batch size**: 16.
- **Learning rate**: 0.001.
- **Optimizer**: Adam.
- **Image size**: 224x224.
- **Loss**: Cross-Entropy.
- **Strategy:** Cross-validation.

Contents

01 Introduction

02 CNNs from scratch

03 Stage 1: Best model

04 Stage 2: Efficiency

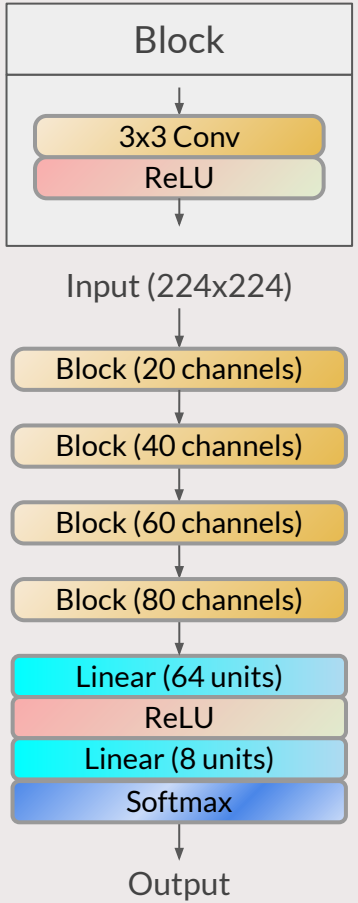
05 Results

06 Conclusions

Our baseline: MCVnet

Hypothesis:

➢ Overparameterized ➡ Overfitting.

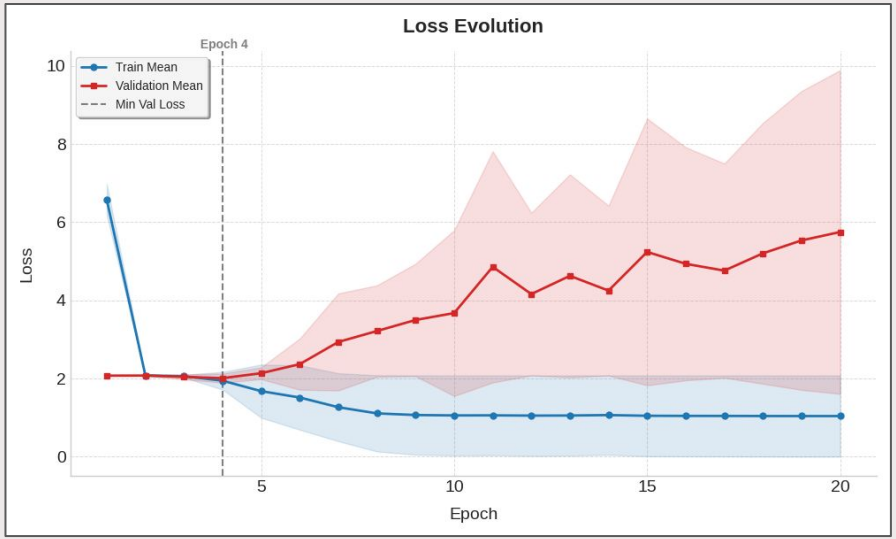


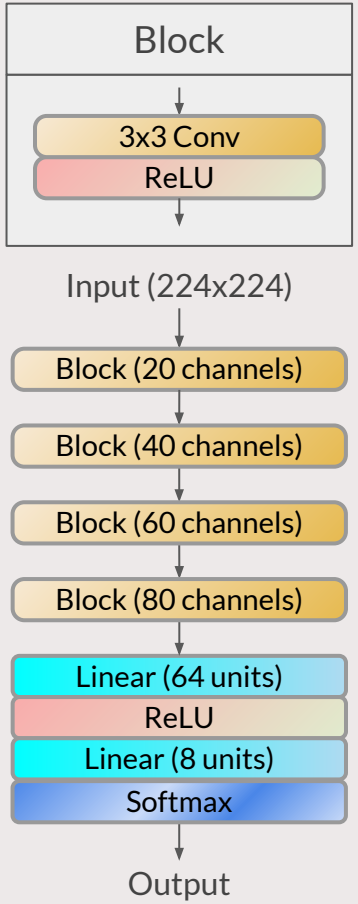
Our baseline: MCVnet

Hypothesis:

➢ Overparameterized ➡ Overfitting.

Parameters	256.97 M
Accuracy	26.8%
Efficiency	1.04e-4
Distance	2569.7

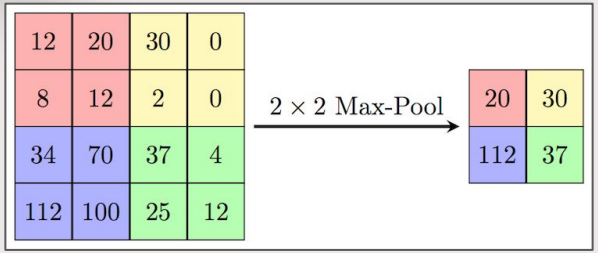


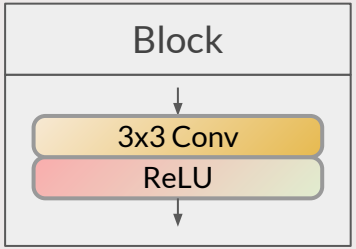


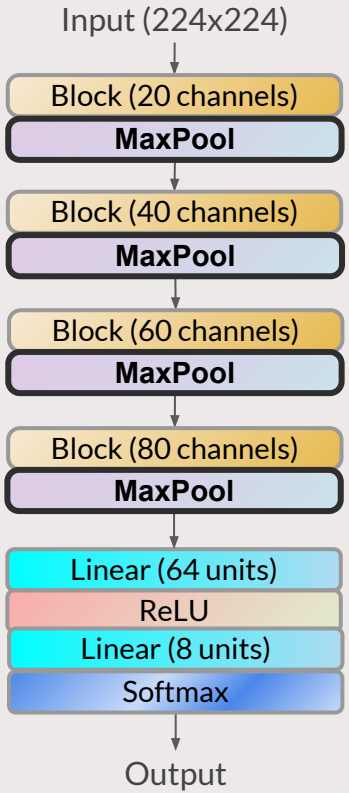
Max-Pool

Hypothesis:

➢ Complexity reduction ➡ Better generalization.



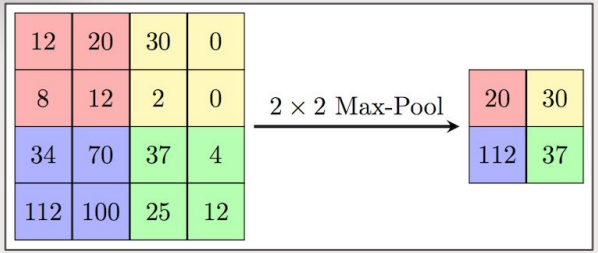


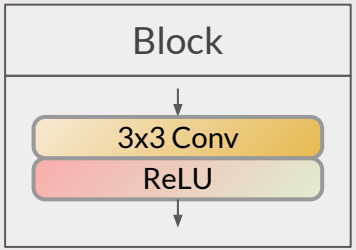


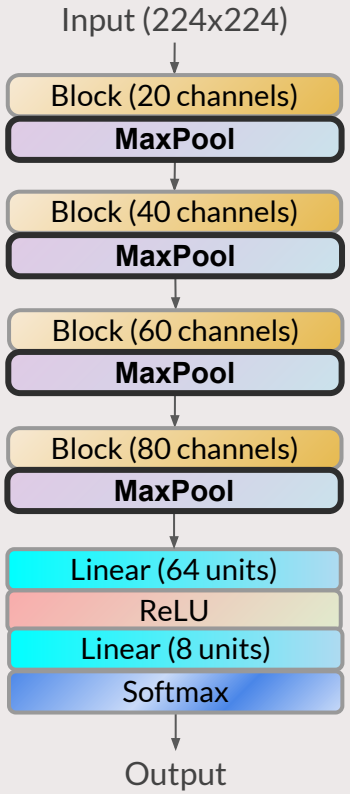
Max-Pool

Hypothesis:

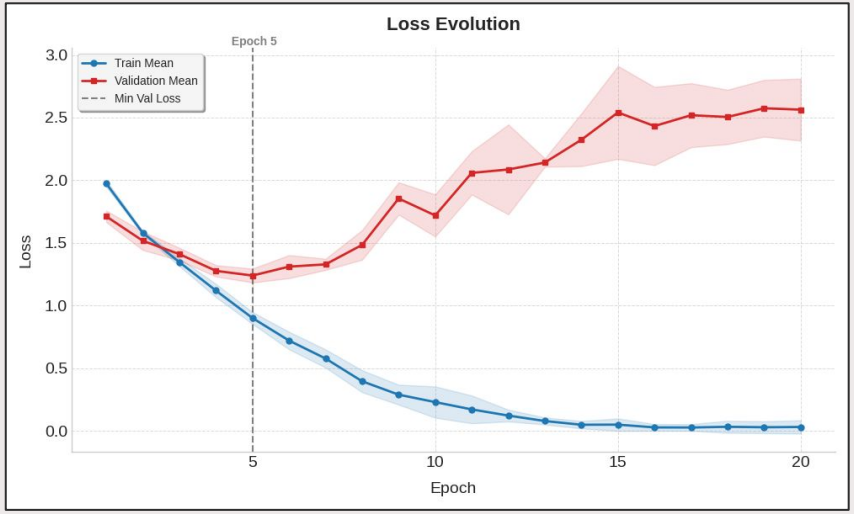
➢ Complexity reduction ➡ Better generalization.



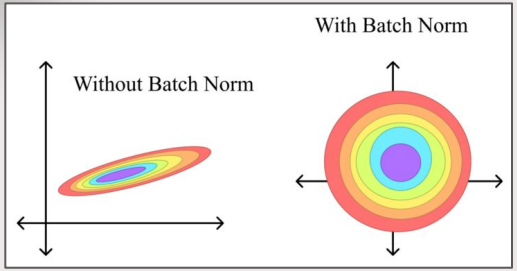




Parameters	1.07 M *(-99.6%)*
Accuracy	57.3% *(+30.5%)*
Efficiency	5.32e-2 *(+51.1%)*
Distance	10.78 *(-99.6%)*

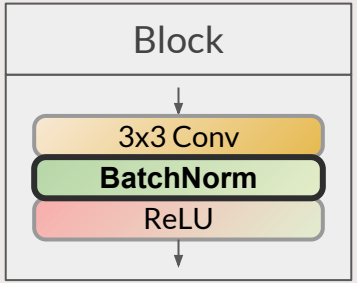


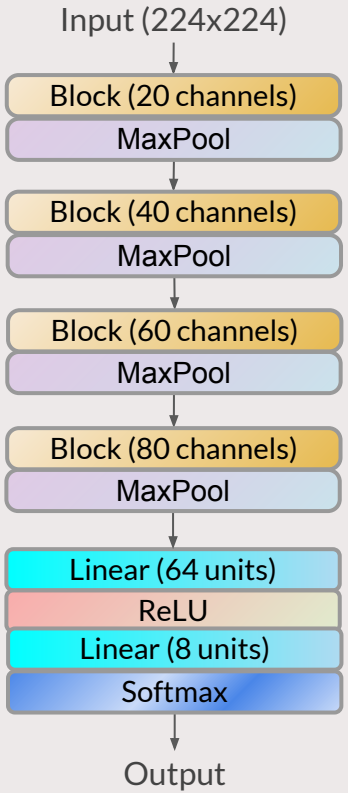
Batch normalization



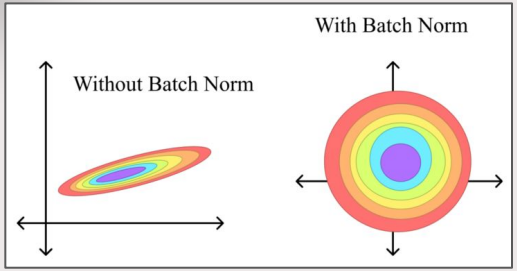
Hypothesis:

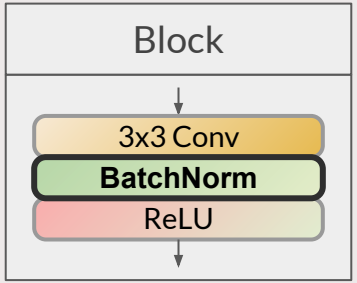
➢ Regularization ➡ Better generalization.

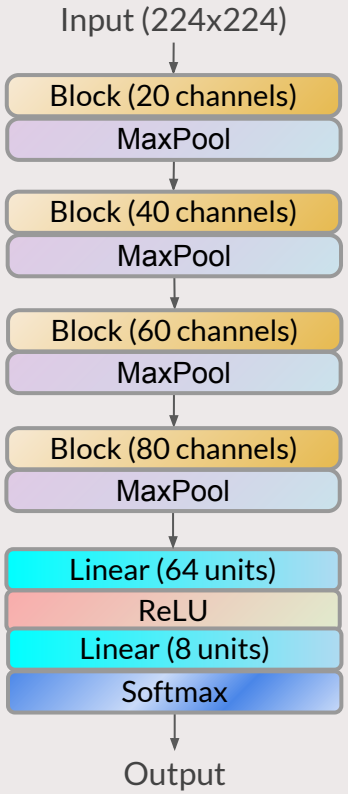




Batch normalization



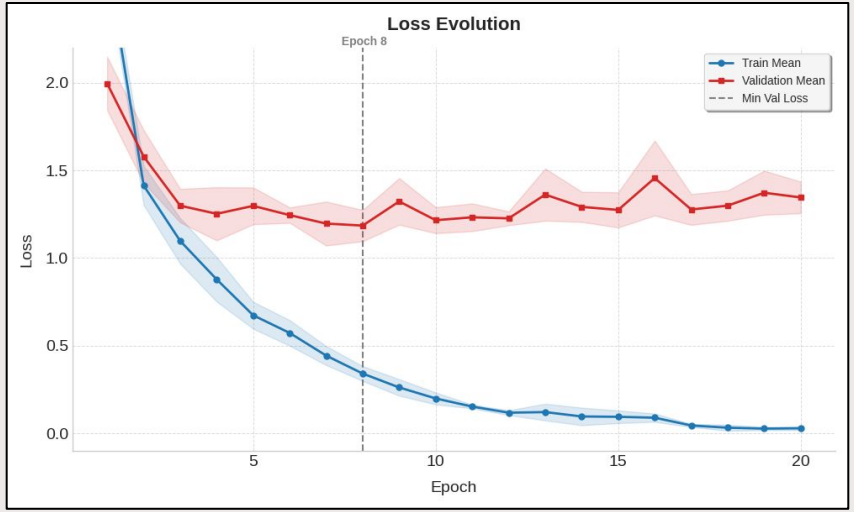




Hypothesis:

➢ Regularization ➡ Better generalization.

Parameters	1.08 M *(≈)*
Accuracy	65.1% *(+7.8%)*
Efficiency	6.04e-2 *(+11.9%)*
Distance	10.78 *(≈)*



Data augmentation

Hypothesis:

➢ The generalization gap is reduced.

1. Random rotation in range (-20, 20) degrees.
2. Random crop within (0.75, 1) of the image area.

Original

Crop[1]

Horiz. Flip (50%)

Rotation[2]

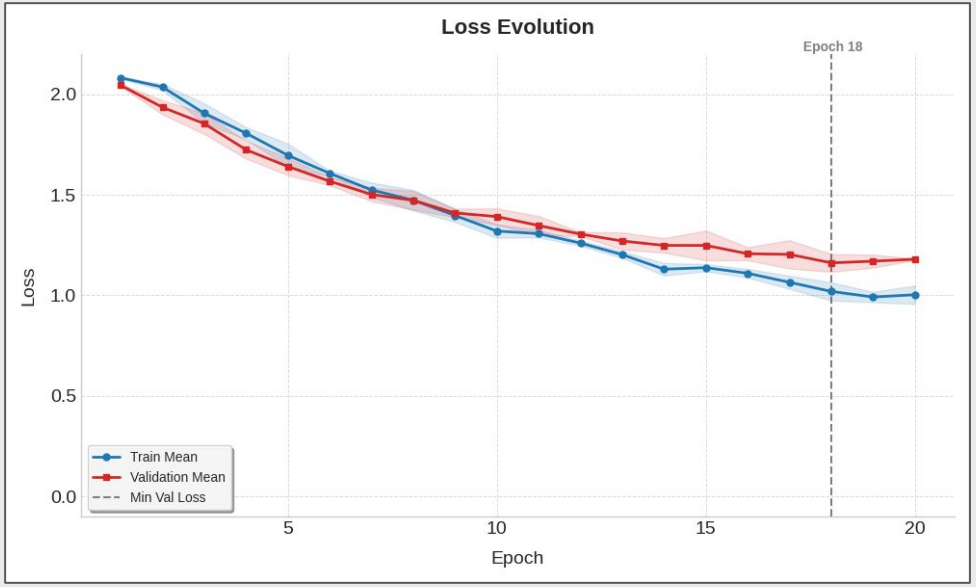
Jittering (30%)

Blur (15%)

Data augmentation

Hypothesis:

➢ The generalization gap is reduced.



1. Random rotation in range (-20, 20) degrees.
2. Random crop within (0.75, 1) of the image area.

Original

Crop[1]

Horiz. Flip (50%)

Rotation[2]

Jittering (30%)

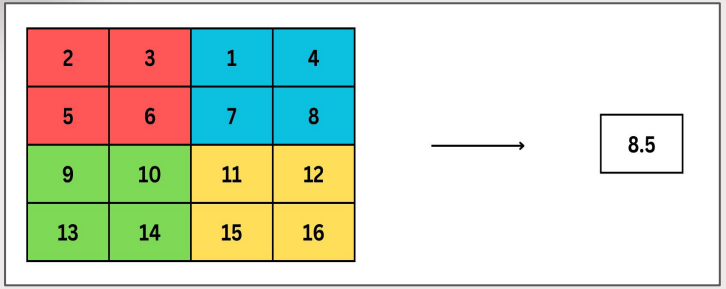
Blur (15%)

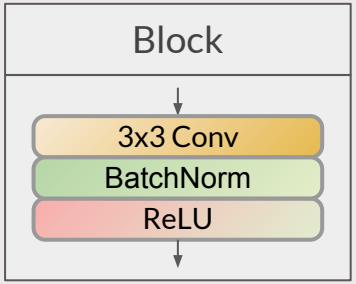
Parameters	1.08 M (≈)
Accuracy	70.1% *(+5%)*
Efficiency	6.51e-2 (≈)
Distance	10.77 (≈)

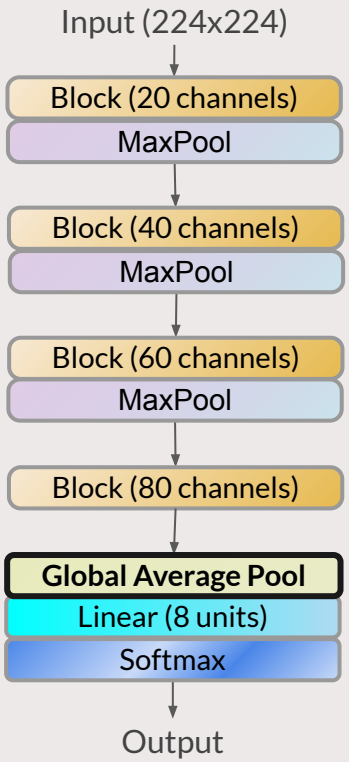
GAP

Hypotheses:

- Complexity reduction ➡ Better generalization.
- Model recognizes more robust and generalized patterns.



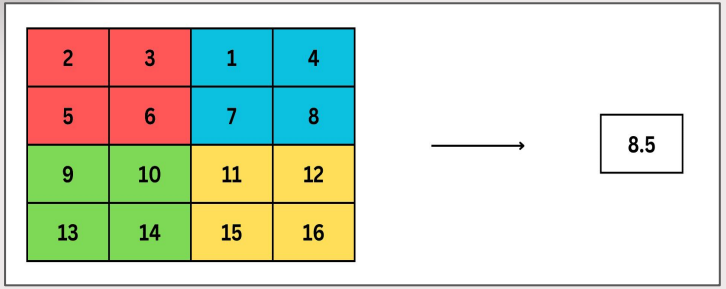


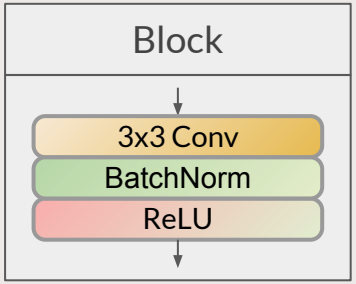


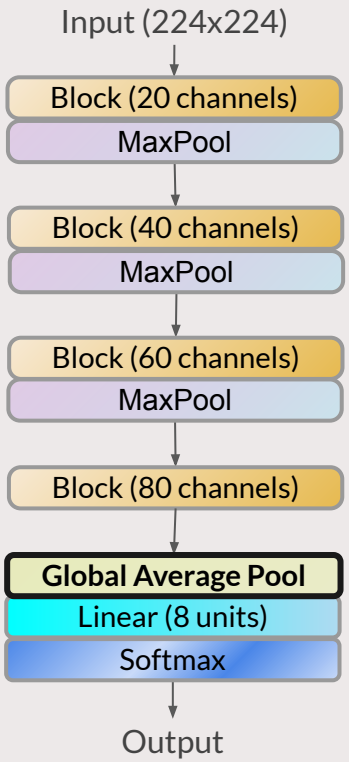
GAP

Hypotheses:

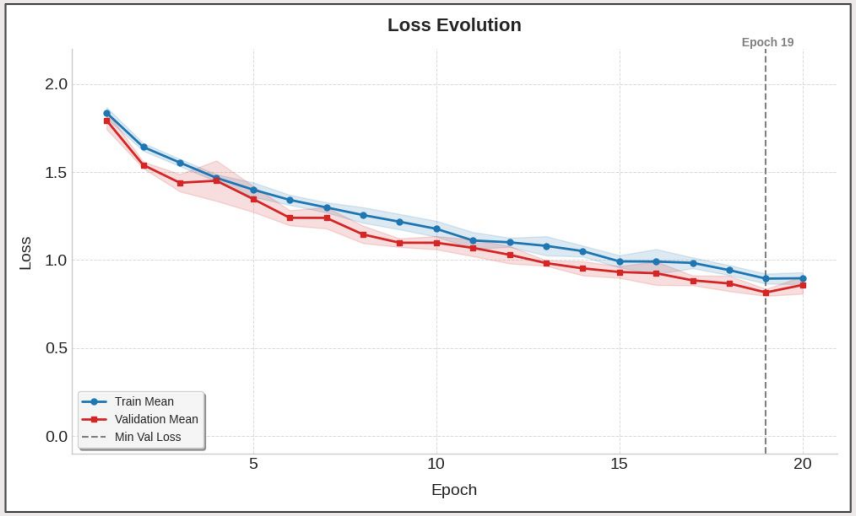
- Complexity reduction ➡ Better generalization.
- Model recognizes more robust and generalized patterns.



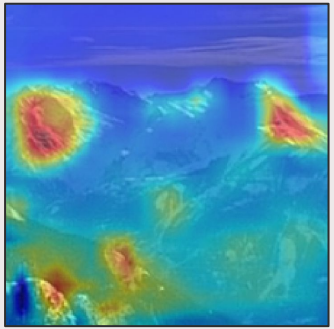
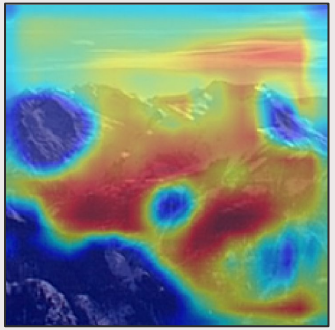
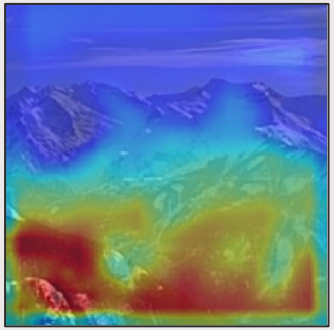
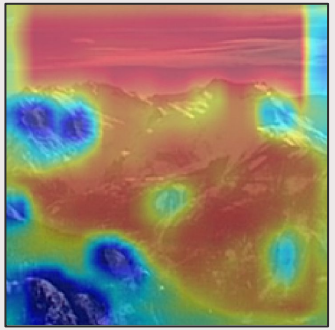




Parameters	73.6 K *(-93.2%)*
Accuracy	72.7% *(+2.6%)*
Efficiency	0.99 *(+1.4%)*
Distance	0.79 *(-92.7%)*



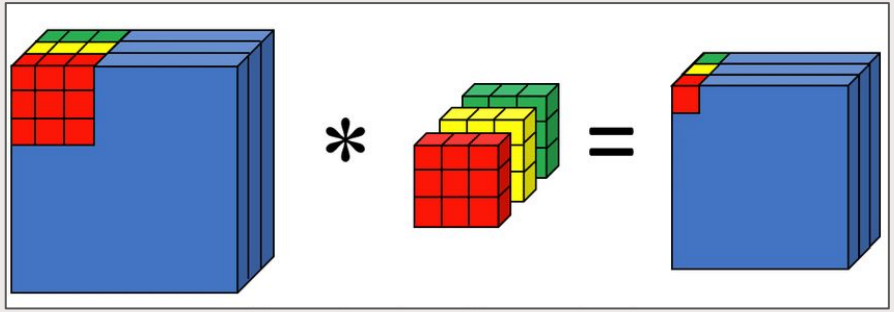
GAP: Grad-CAM

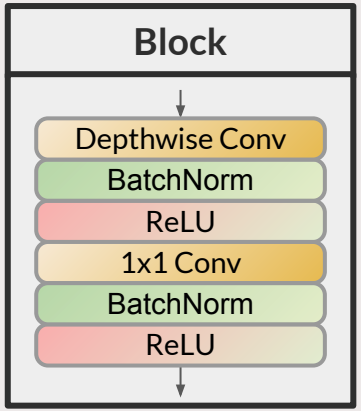
		Mountain class neuron	**Coast** class neuron
GAP	**Pred:** Mountain		
w/o GAP	**Pred:** Coast		

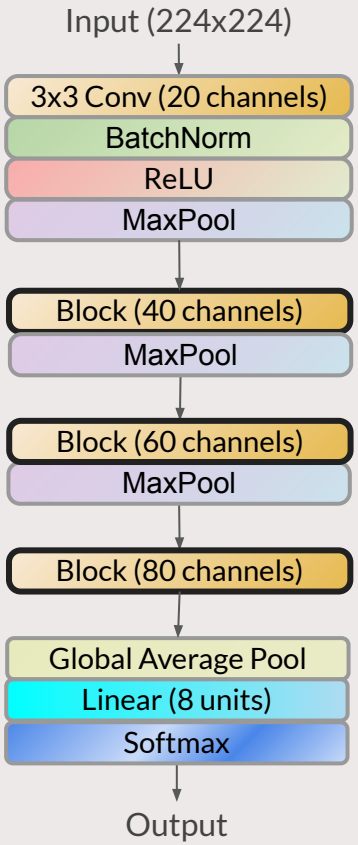
Depthwise convolutions

Hypothesis:

➢ Accuracy is maintained with fewer parameters.



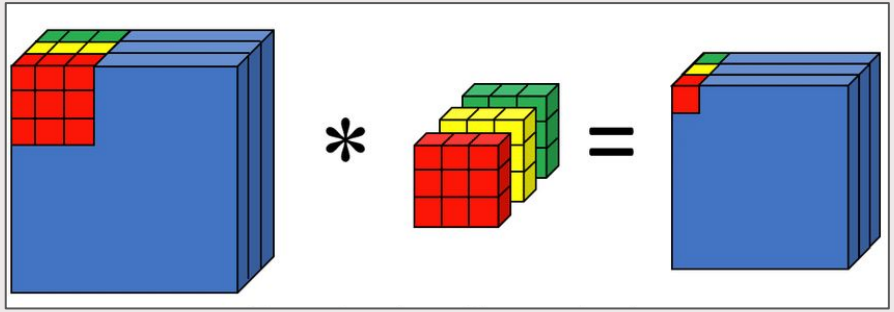




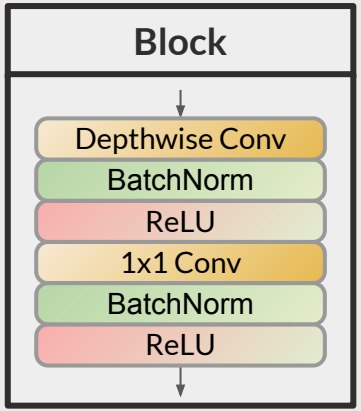
Depthwise convolutions

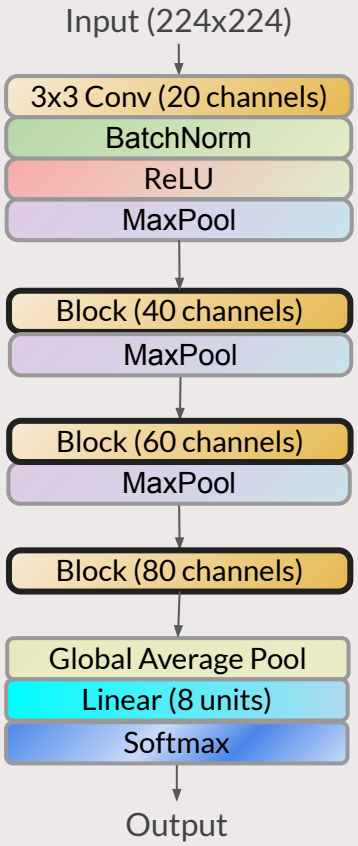
Hypothesis:

➢ Accuracy is maintained with fewer parameters.



Parameters	10.9 K *(-85.2%)*
Accuracy	74.2% *(+1.5%)*
Efficiency	6.8 *(586%)*
Distance	0.28 *(-65%)*

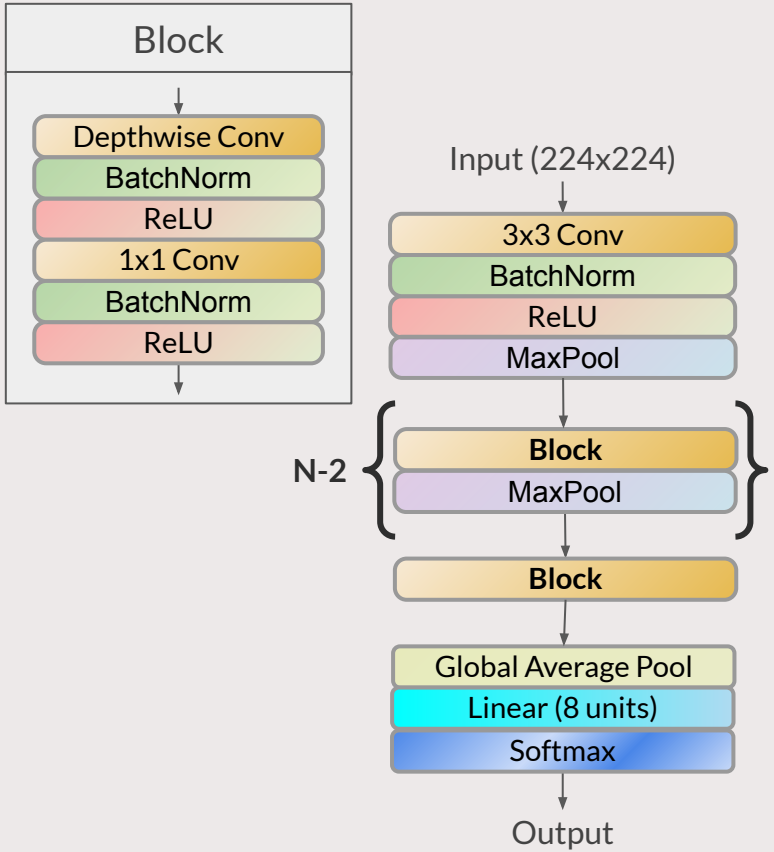




Number of blocks

Hypothesis:

➢ Tradeoff performance ⟷ nº parameters.



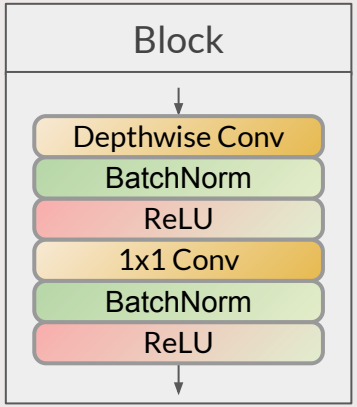
Number of blocks

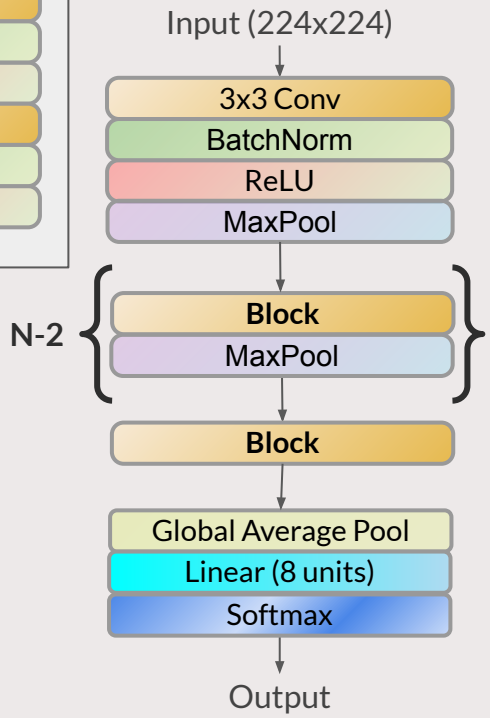
Hypothesis:

➢ Tradeoff performance ⟷ nº parameters.

Blocks (N)	Parameters (K)	Accuracy (%)	Efficiency	Distance
2	**2.0**	52.8	**26.3**	0.47
3	5.1	62.4	12.2	0.38
4	10.9	74.2	6.8	**0.28**
5	20.1	78.9	3.9	0.29
6	33.6	**82.3**	2.4	0.38

Last layer ➡ 14x14 feature maps.

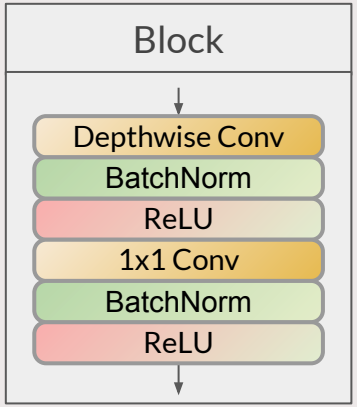


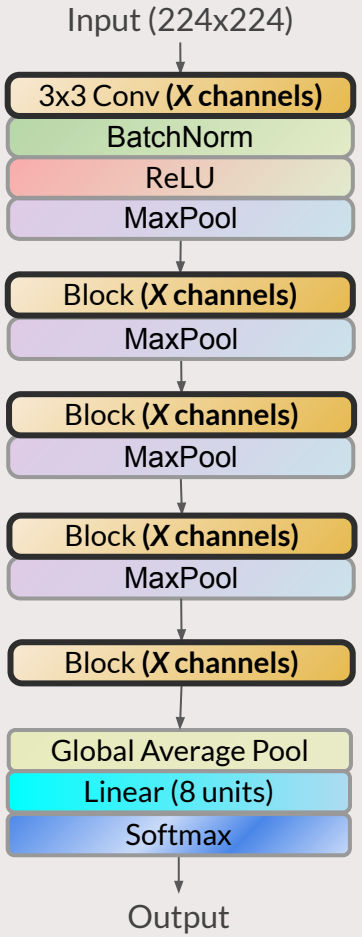


Number of channels

Hypotheses:

- Tradeoff performance ⟷ nº parameters.
- Pyramidal expansion leads to better performance.



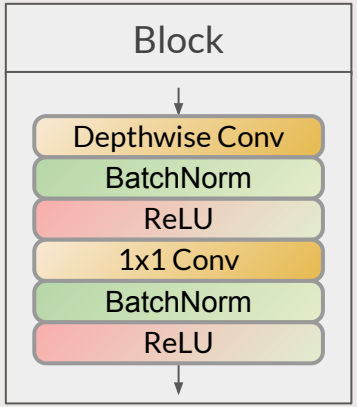


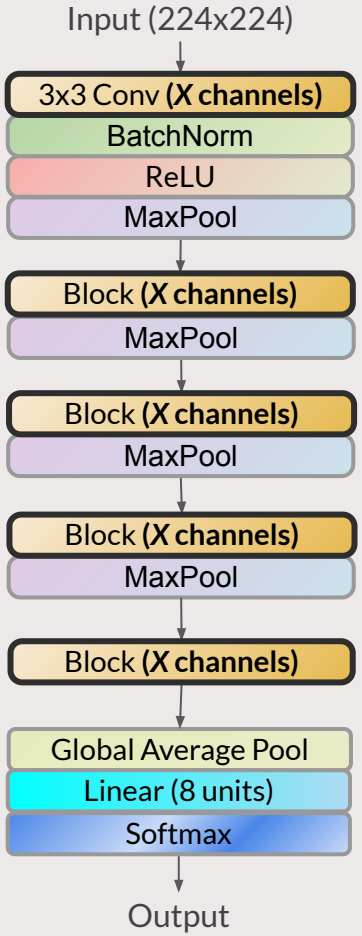
Number of channels

Hypotheses:

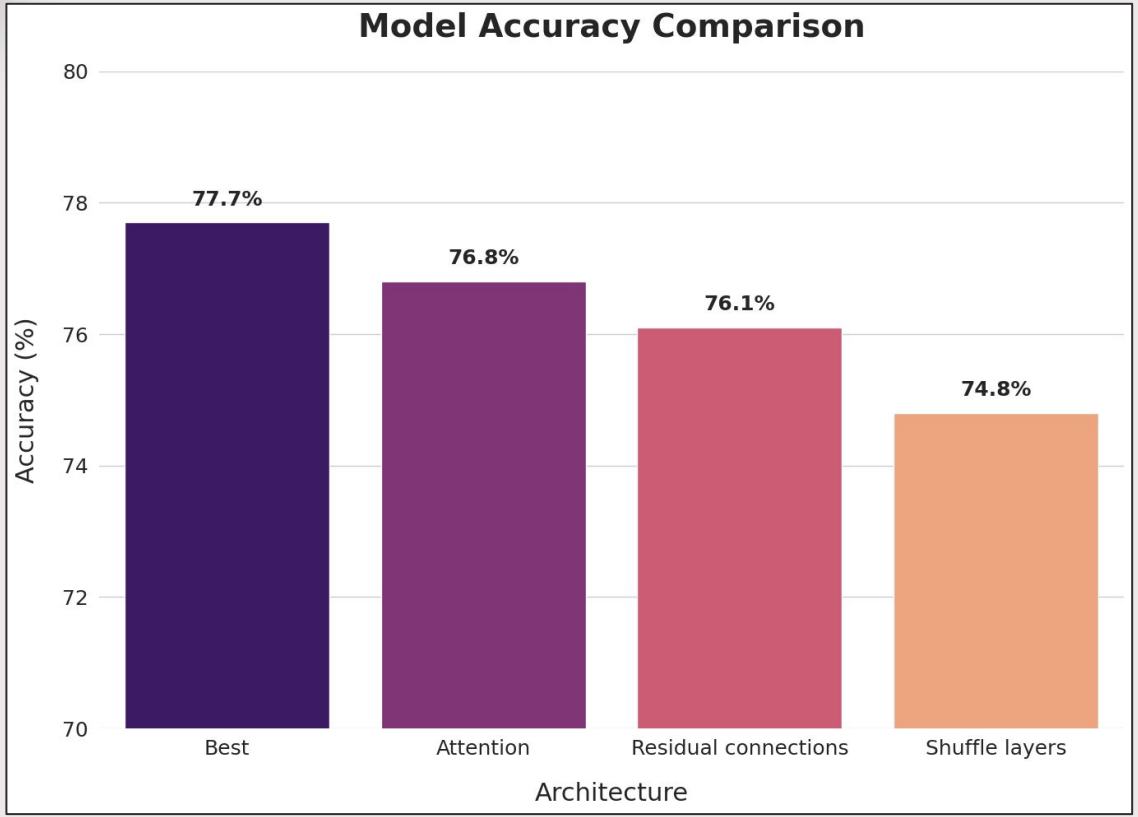
- Tradeoff performance ⟷ nº parameters.
- Pyramidal expansion leads to better performance.

Channels	Parameters (K)	Accuracy (%)	Efficiency	Distance
[5, 10, 15, 20, 25]	**2.0**	60.4	**29.6**	0.40
[20, 20, 20, 20, 20]	3.4	67.1	19.8	0.33
[10, 20, 30, 40, 50]	6.1	72.4	11.9	0.28
[45, 45, 45, 45, 45]	12.1	76.8	6.3	0.26
[15, 30, 45, 60, 75]	12.1	77.7	6.4	**0.25**
[20, 40, 60, 80, 100]	20.1	**78.9**	3.9	0.29





What did not work?

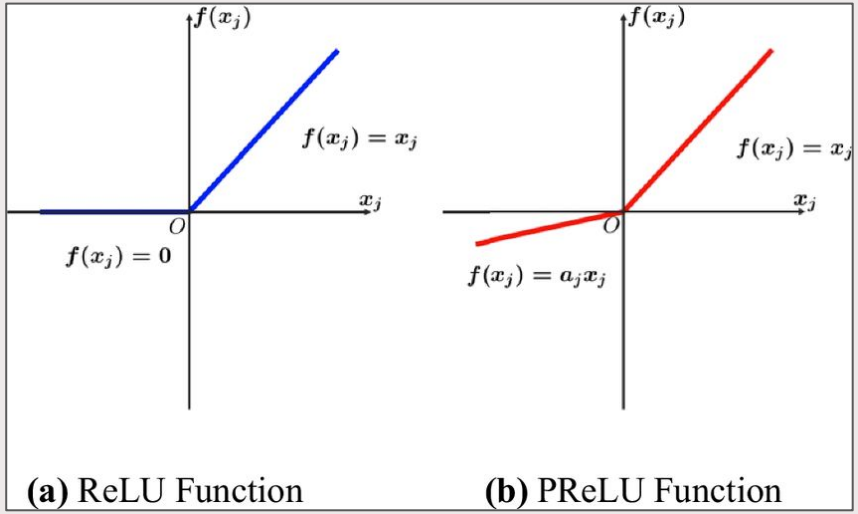


	Parameters
Best	12.1 K
Attention	17.1 K
Residual connections	21.6 K
Shuffle layers	8.5 K

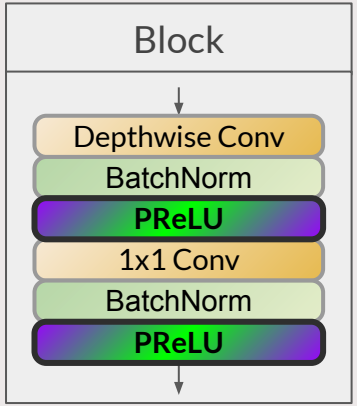
Activations

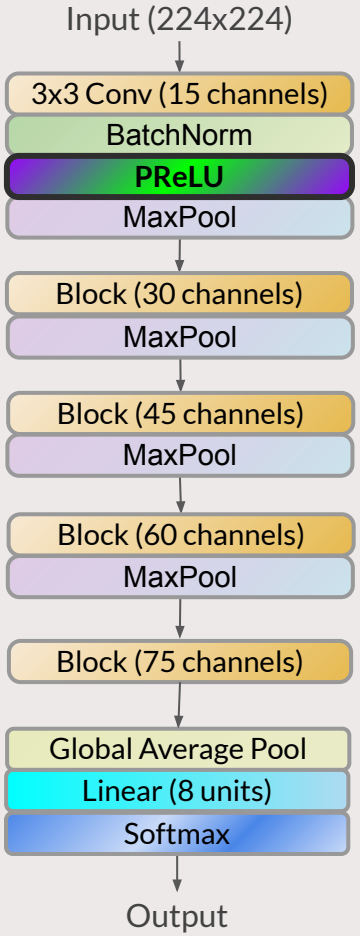
Hypothesis:

➢ Performance is slightly improved.



(a) ReLU Function **(b)** PReLU Function

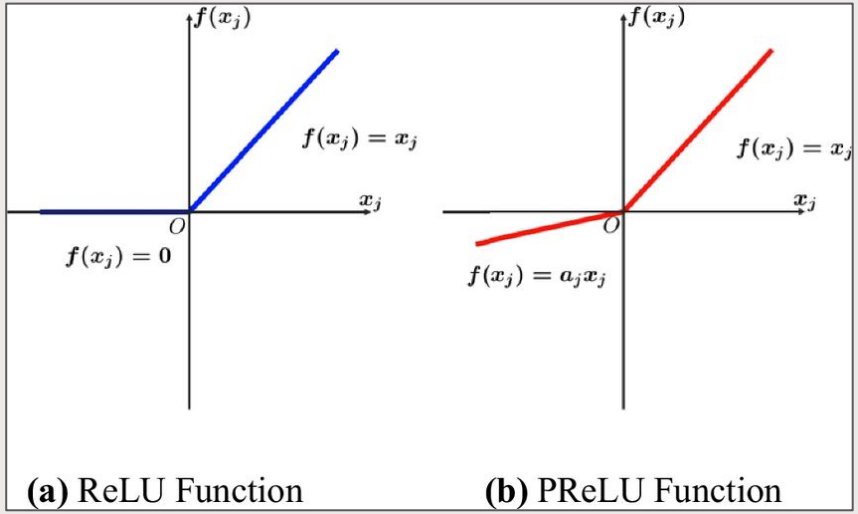




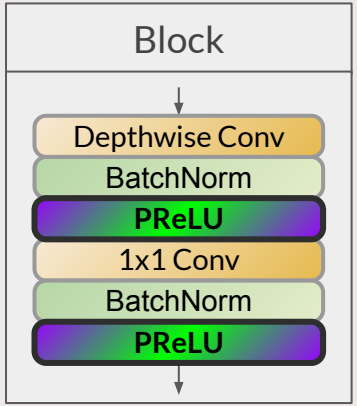
Activations

Hypothesis:

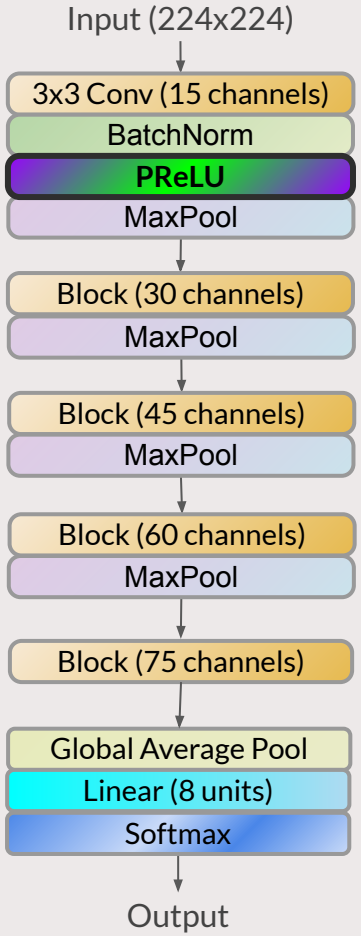
➢ Performance is slightly improved.



(a) ReLU Function **(b)** PReLU Function



Parameters	12.1 K (≈)
Accuracy	78.3% (+2.6%)
Efficiency	6.5 (+1.5%)
Distance	0.25 (≈)



Hyperparameter search

We fix the architecture to optimize the configuration.

Full Random Search Space:

Batch Size: [8, 16, 32, 64]
Dropout 1: [0.0, 0.1, 0.2, 0.3]
Dropout 2: [0.0, 0.2, 0.3, 0.5]
Learning Rate: [1e-4, 1e-3, 1e-2, 0.1]
Momentum: [0.0, 0.5, 0.9]
Epochs: [20, 50, 100]
Optimizer: [SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, Nadam]
Weight Decay: [0.0, 1e-6, 1e-5, 1e-4, 1e-3]

Stage 1 - Random Search: 20 trials

- The Top 5 performing configurations are selected to initialize the next stage.

	Batch Size	Dropout 1	Dropout 2	Learning Rate	Momentum	Epochs	Optimizer	Weight Decay	Distance
Top 1	16	0.1	0.5	0.01	0.5	100	Adamax	1e-3	0.2224
Top 2	64	0.0	0.0	0.001	0.0	50	Nadam	1e-6	0.2321
Top 3	32	0.1	0.3	0.001	0.5	50	RMSprop	1e-5	0.2439
Top 4	8	0.2	0.5	0.01	0.9	100	Adamax	0.0	0.2591
Top 5	64	0.2	0.2	0.001	0.9	100	RMSprop	1e-4	0.2952

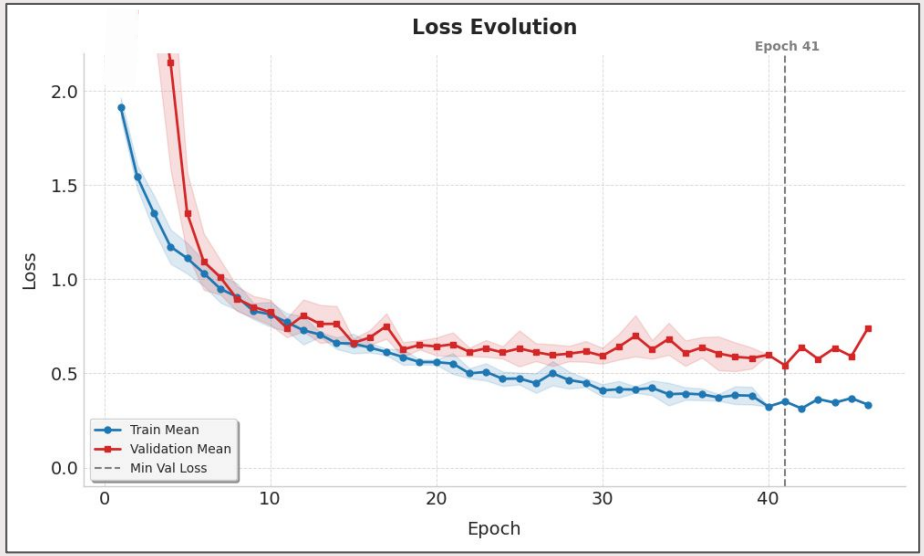
Hyperparameter search

Stage 2 - Bayesian Optimization: 40 trials

Top configuration:

Batch Size	Dropout 1	Dropout 2	Learning Rate	Momentum	Epochs	Optimizer	Weight Decay
64	0.003	0.389	7.96e-4	0.758	50	RMSprop	1.20e-6

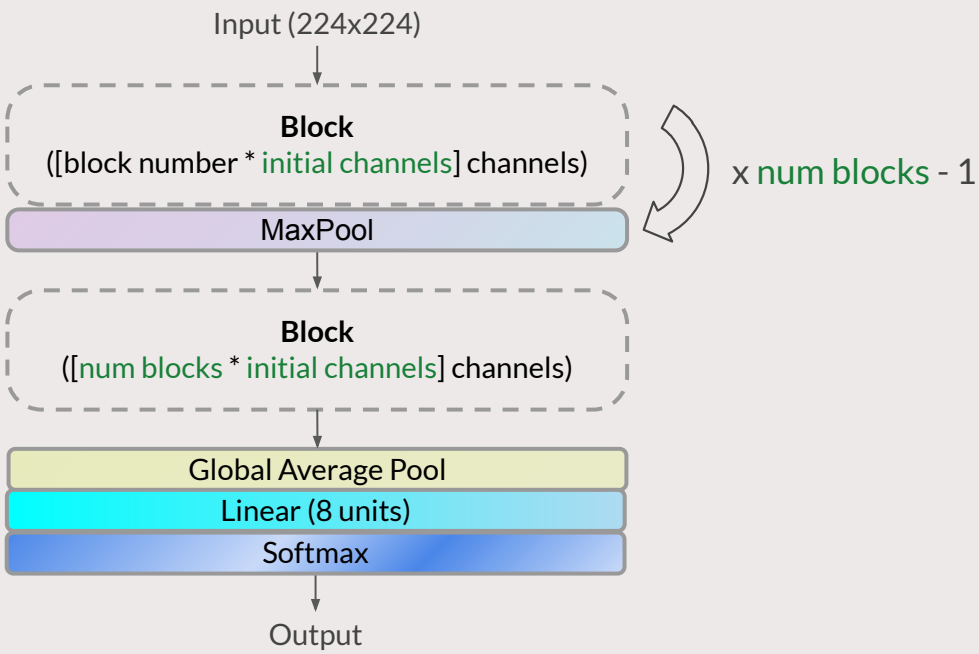
Parameters	12.1 K *(≈)*
Accuracy	82.2% *(+3.9%)*
Efficiency	6.78 *(+4.3%)*
Distance	0.22 *(-12%)*

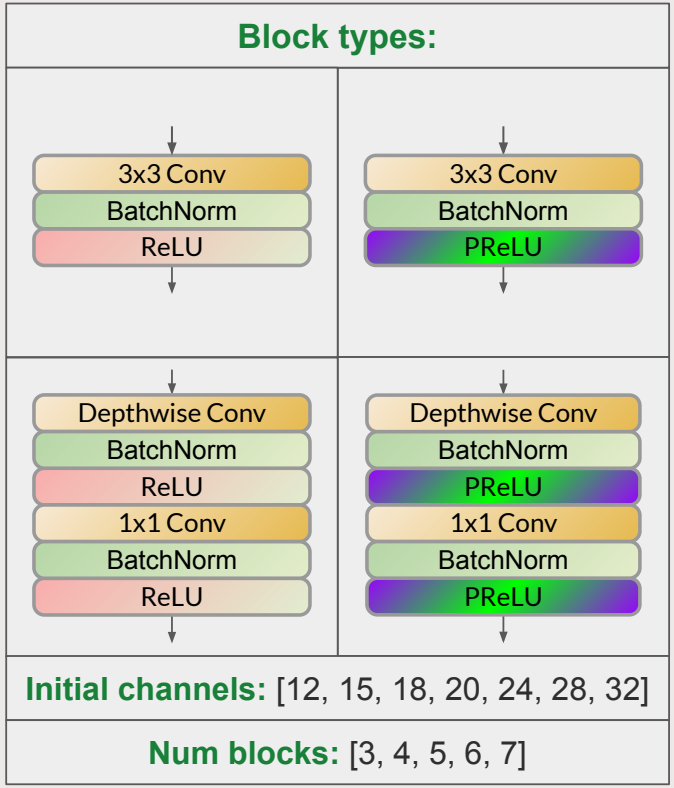


Neural Architecture Search (NAS)

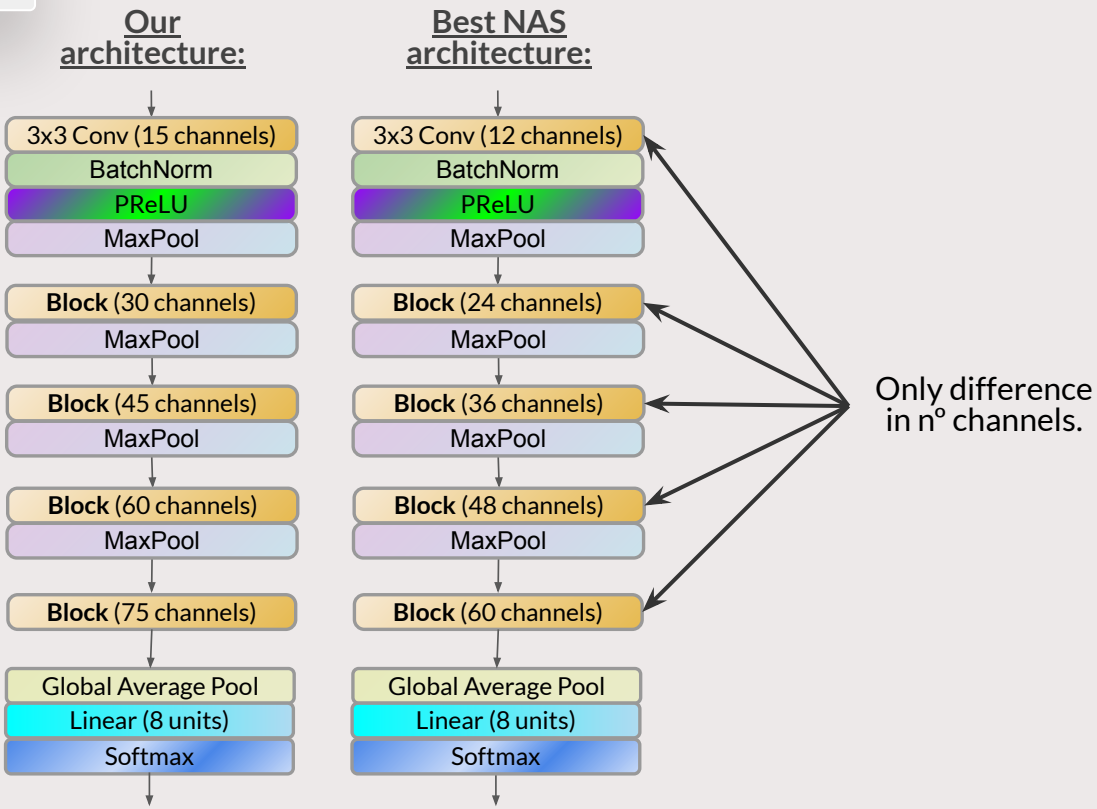
The model learns its optimal architecture over 40 trials.

Objective: Minimize distance.





Neural Architecture Search (NAS)



Parameters	8.26 K *(-31.7%)*
Accuracy	81.5% *(-0.7%)*
Efficiency	9.86 *(+45.4%)*
Distance	0.20 *(-9.1%)*

Contents

01 Introduction

02 CNNs from scratch

03 Stage 1: Best model

04 Stage 2: Efficiency

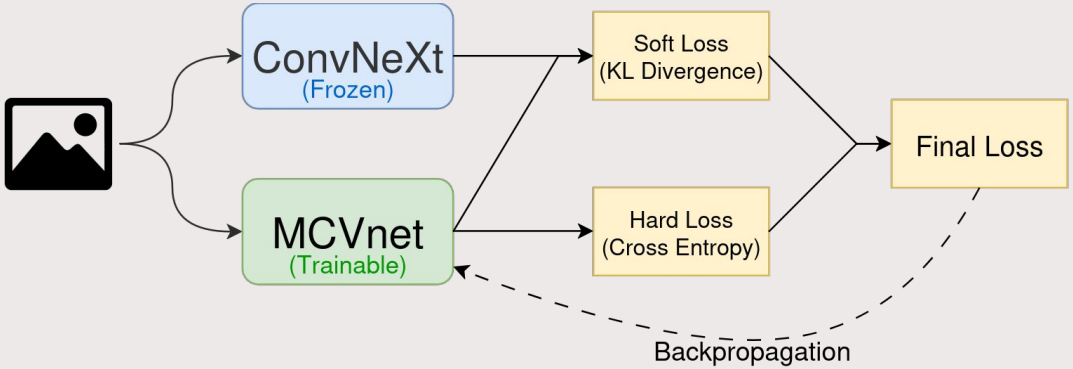
05 Results

06 Conclusions

Distillation

Hypothesis:

➢ With ConvNeXt as teacher we get better results.



$$L_{\text{final}} = (1-\alpha)\cdot t^2 \cdot L_{\text{kd}} + \alpha \cdot L_{\text{task}} \quad [1]$$

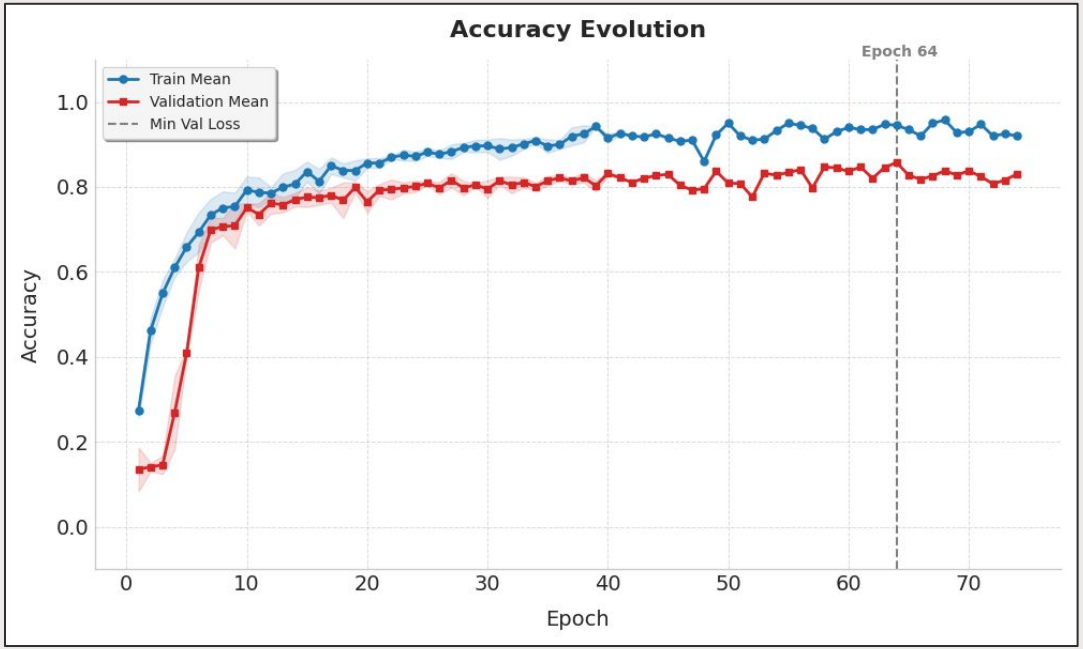
[1] Ahmadabadi et al., *Distilling Knowledge from CNN-Transformer Models for Enhanced Human Action Recognition.*

Distillation

Hypothesis:

➢ With ConvNeXt as teacher we get better results.

Parameters	8.26 K *(=)*
Accuracy	83.3% *(+1.8%)*
Efficiency	10.09 *(+2.3%)*
Distance	0.19 *(-5.1%)*



Pruning

Hypothesis:

➢ Remove unnecessary weights ➡ Greater efficiency.

Note: Pruning performed using pretrained weights on best fold.

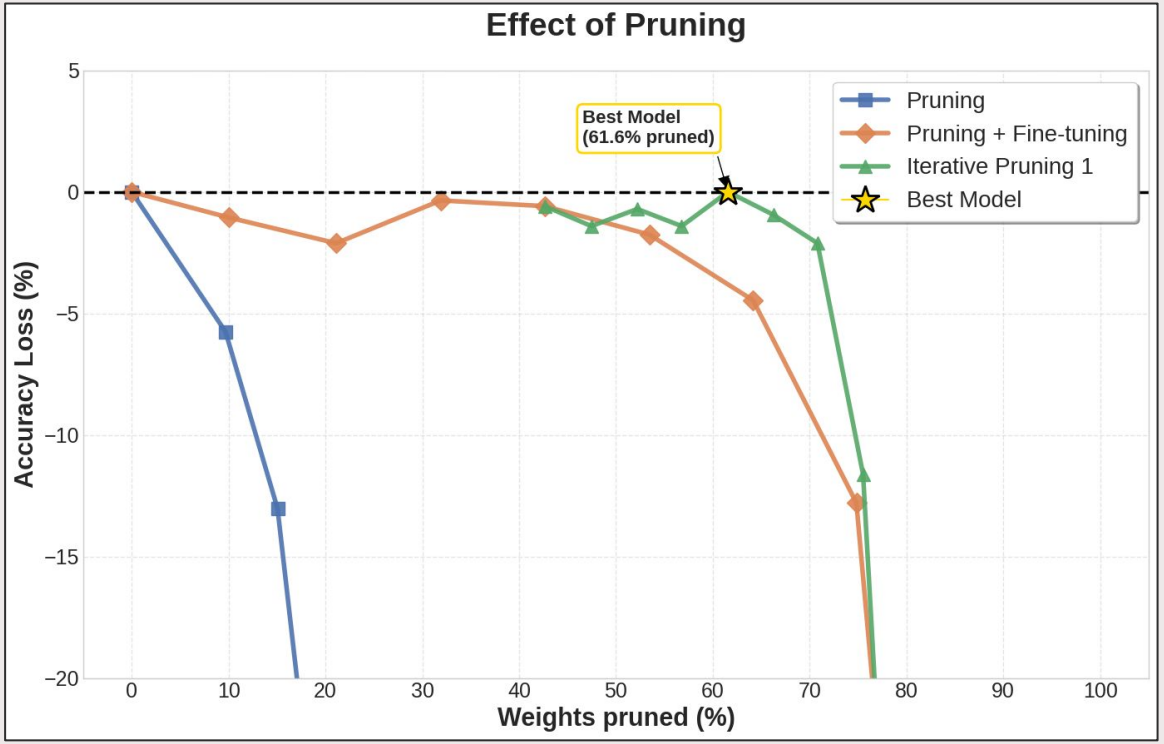
Pruning

Note: Pruning performed using pretrained weights on best fold.

	Fold 3	Pruned
Parameters	8.26 K	**3.18 K**
Accuracy	**85.3%**	85.2%
Efficiency	10.32	**26.8**
Distance	0.17	**0.15**

Hypothesis:

➢ Remove unnecessary weights ➡ Greater efficiency.



Contents

01 Introduction

02 CNNs from scratch

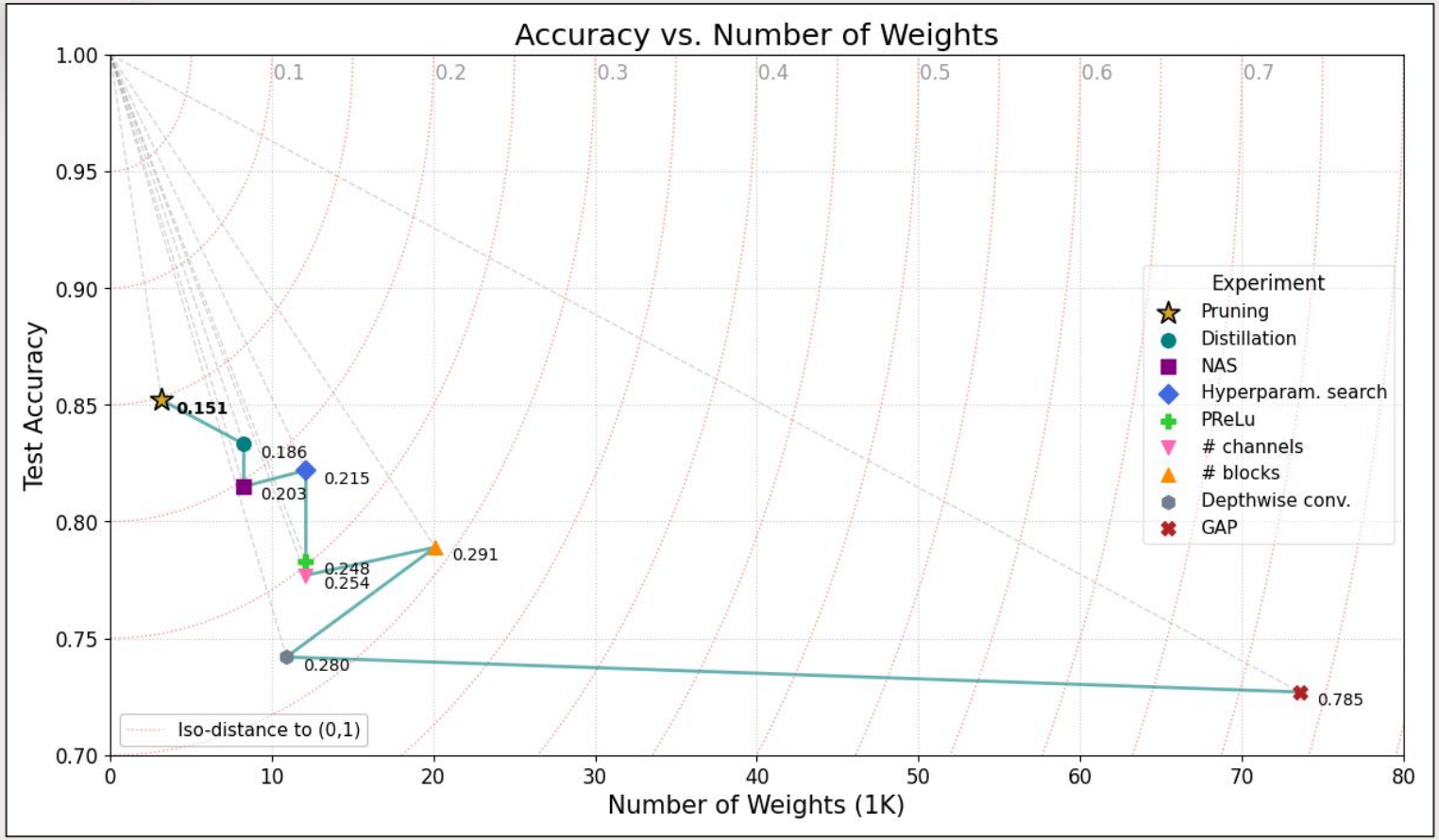
03 Stage 1: Best model

04 Stage 2: Efficiency

05 Results

06 Conclusions

Final results



	Distance
Week 3	42.50
Baseline	2569.7
Max pool	10.78
Batchnorm	10.78
Data aug.	10.77

Contents

01 Introduction

02 CNNs from scratch

03 Stage 1: Best model

04 Stage 2: Efficiency

05 Results

06 Conclusions

Conclusions

Week 1

We explored the strengths and limitations of traditional handcrafted techniques.

Week 2

We tested if MLPs by themselves could outperform traditional methodologies.

Week 3

We adapted a well-known CNN architecture (MobileNetV3) to our specific case.

Week 4

We built and trained an efficient CNN from scratch.

MCV MASTER IN COMPUTER VISION Barcelona

IMAGE CLASSIFICATION

Task 4: Convolutional Networks from scratch

Group 5

Marc Artero · Diego Hernández · Bernat Medina · Pau Monserrat

Module C3: Machine Learning for Computer Vision