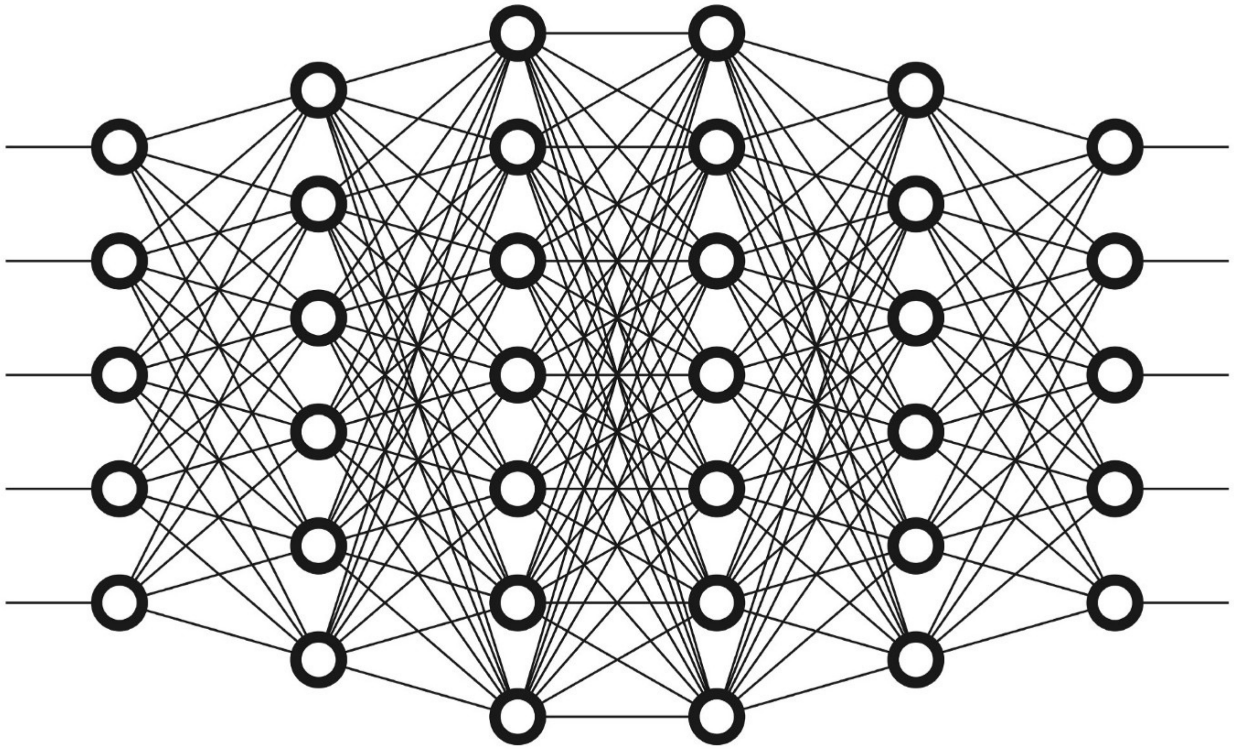
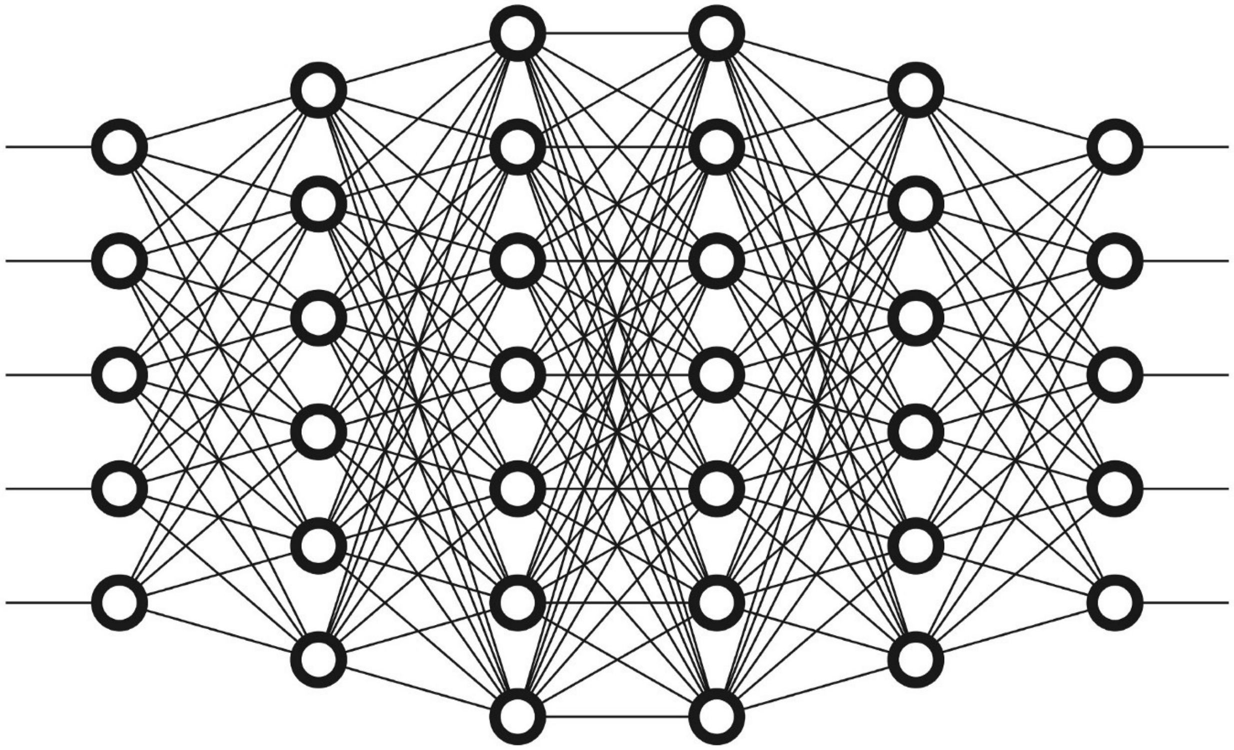
STRATEGIES FOR TRAINING DEEP LEARNING



OLARIK SURINTA

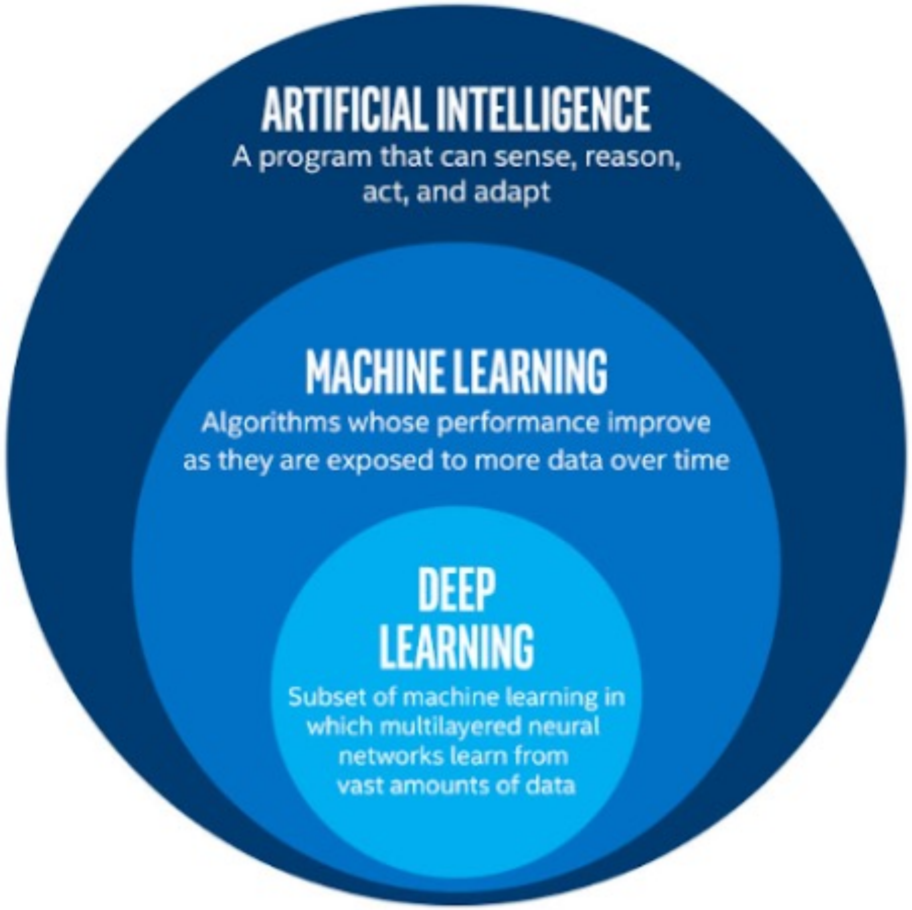
Department of Information Technology, Faculty of Informatics, Mahasarakham University, Thailand

OUTLINE

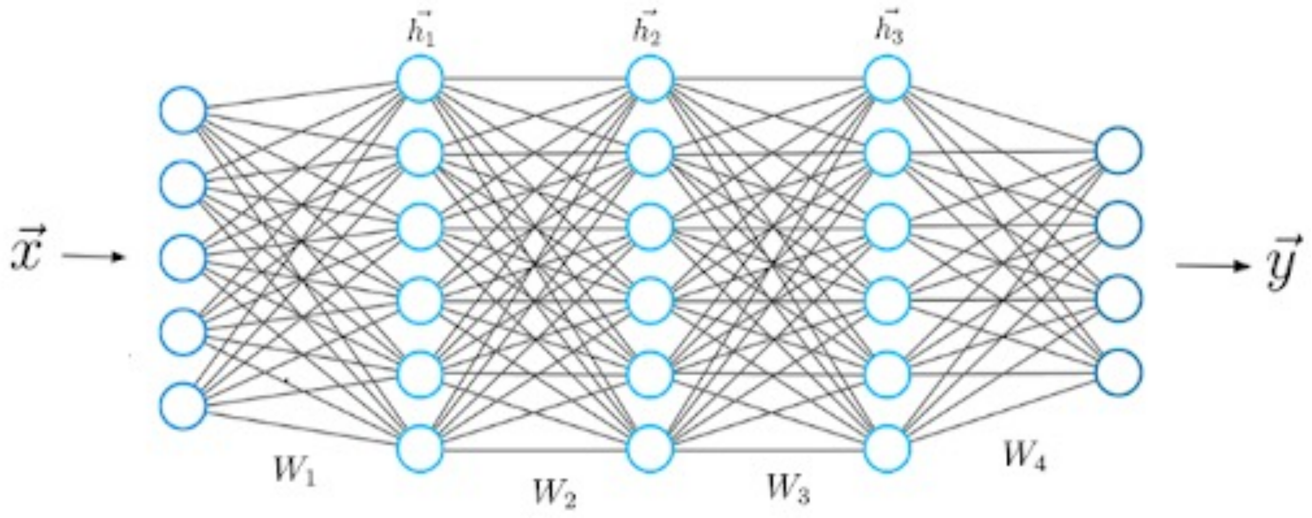


- INTRODUCTION TO DEEP LEARNING
- TRAIN DEEP LEARNING MODEL
- SCRATCH AND TRANSFER LEARNING
- HYPERPARAMETERS TUNING
- DATA AUGMENTATION
- ENSEMBLE LEARNING
- FUSION CNN
- SNAPSHOT ENSEMBLE CNN

INTRODUCTION TO DEEP LEARNING



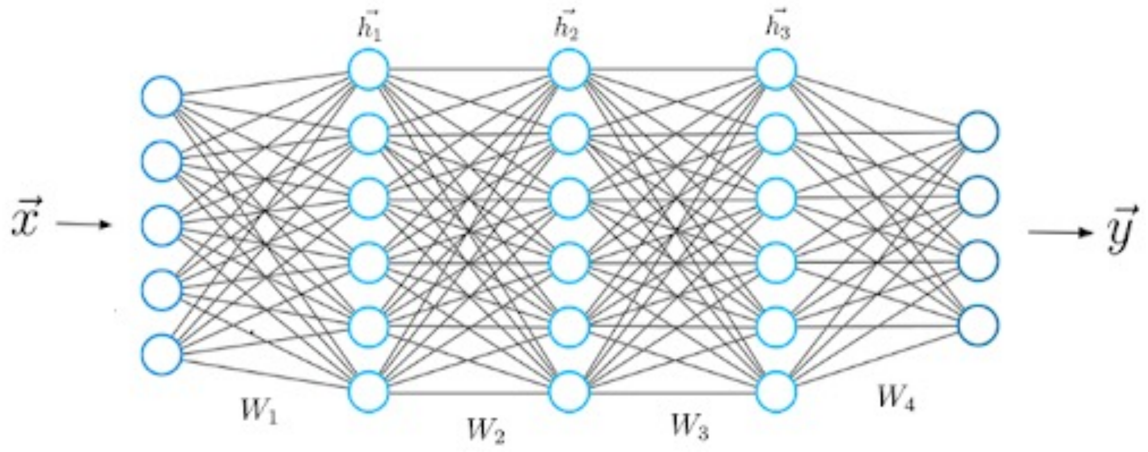
Modern deep learning models are designed based on artificial neural networks.

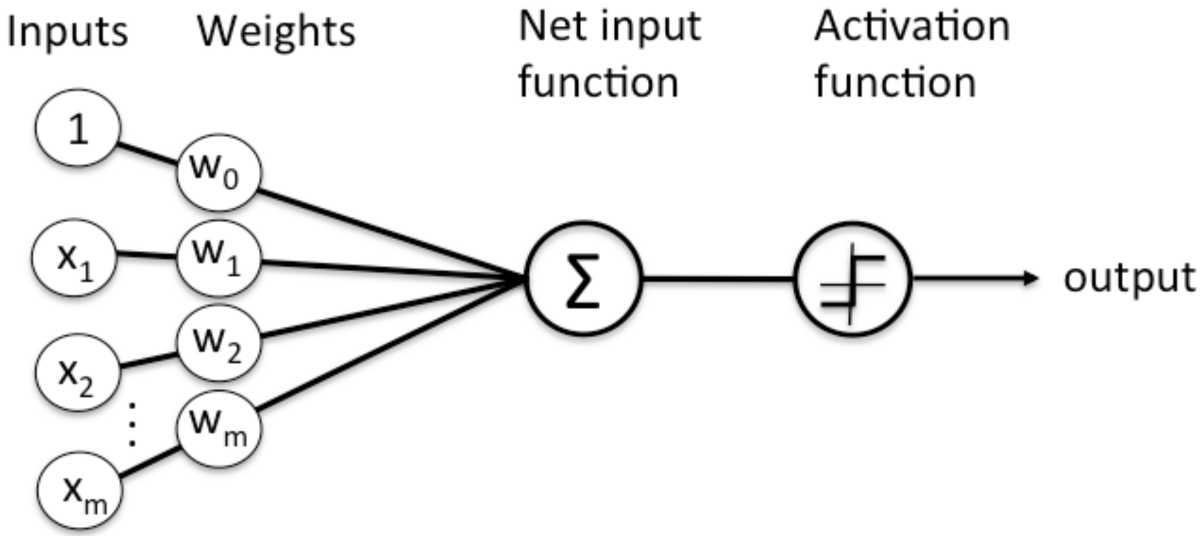


https://builtin.com/machine-learning/what-is-deep-learning

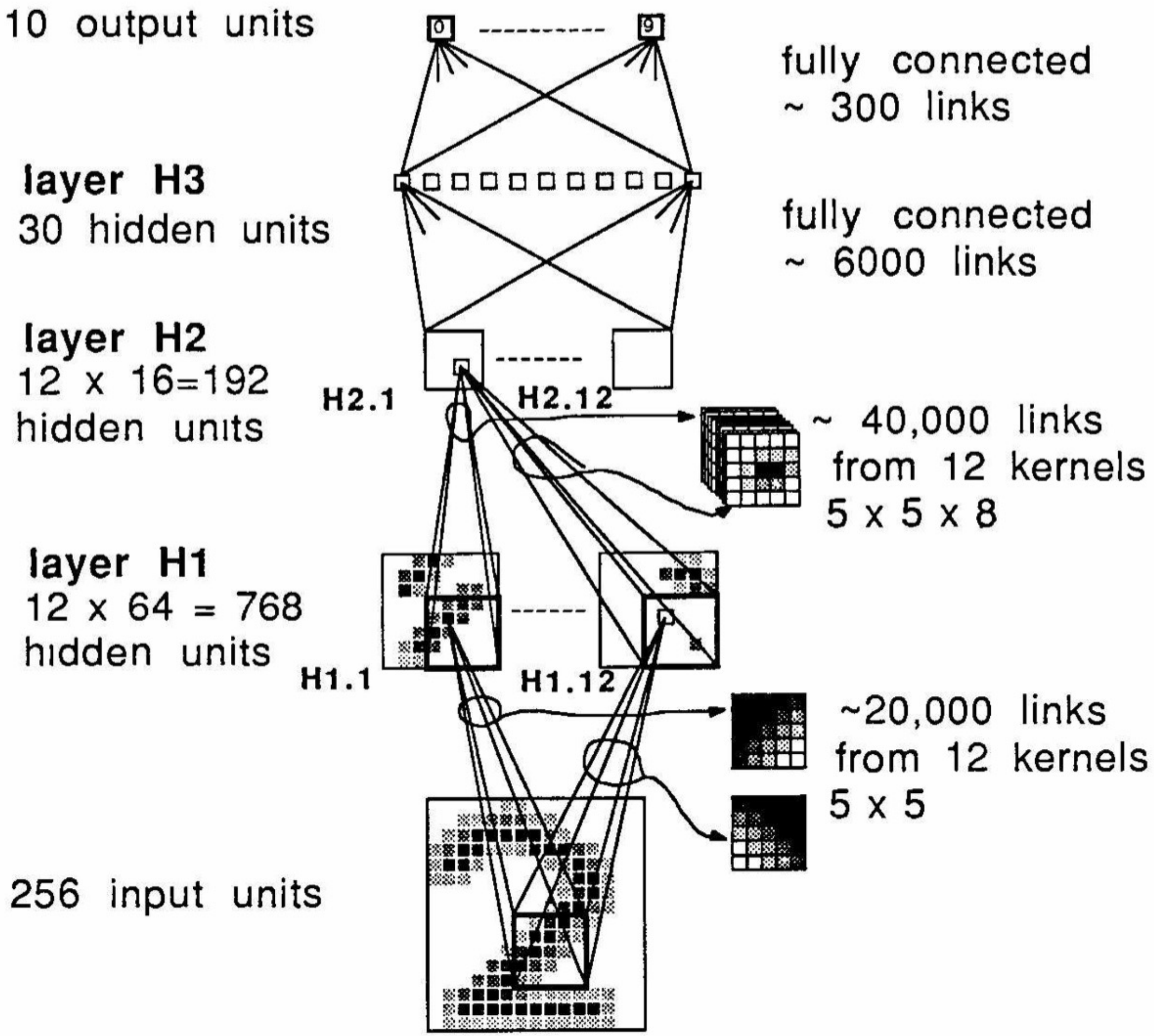
INTRODUCTION TO DEEP LEARNING

Neural Network





https://wiki.pathmind.com/neural-network



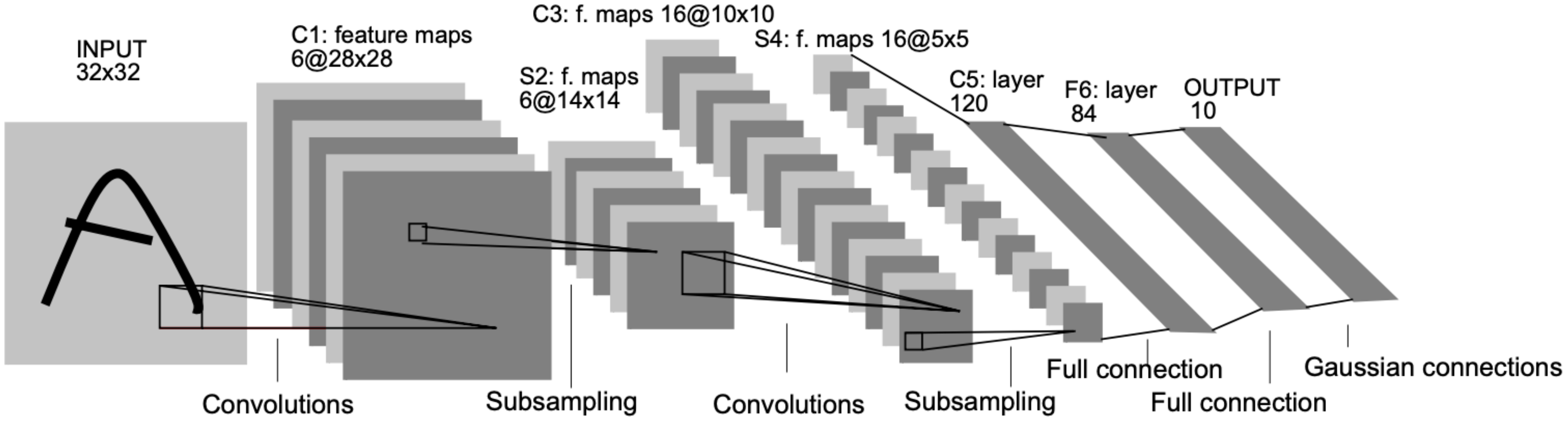
INTRODUCTION TO DEEP LEARNING

LeCun *et al.* (1989)
"A single network learns the entire recognition operation, going from the normalized image of character to the final classification."

LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," in *Neural Computation*, vol. 1, no. 4, pp. 541-551, Dec. 1989, doi: 10.1162/neco.1989.1.4.541.

INTRODUCTION TO DEEP LEARNING

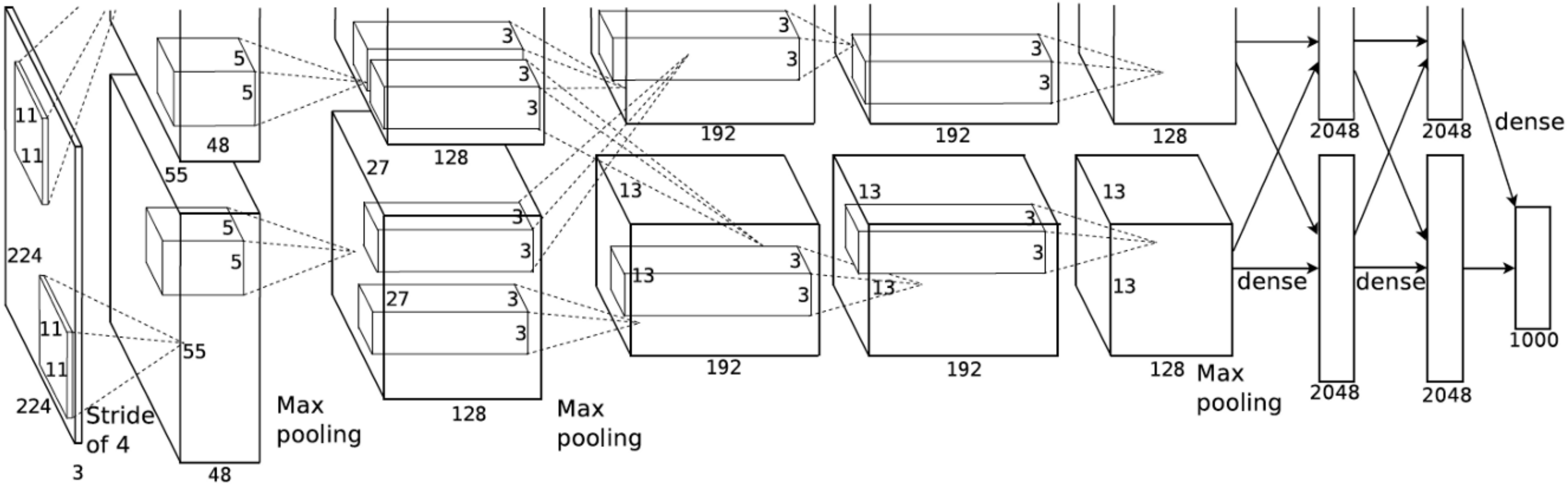
LeCun et al. (1998) proposed a convolutional neural network called LeNet5.



LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. 1998. Gradient-Based Learning Applied to Document Recognition. IEEE. 86, 11 (1998), 2278–2324.

INTRODUCTION TO DEEP LEARNING

AlexNet



Krizhevsky, A., Sutskever, I., Hinton, G. 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25 (NIPS 2012)

INTRODUCTION TO DEEP LEARNING

ILSVRC. There are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images.

AlexNet

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

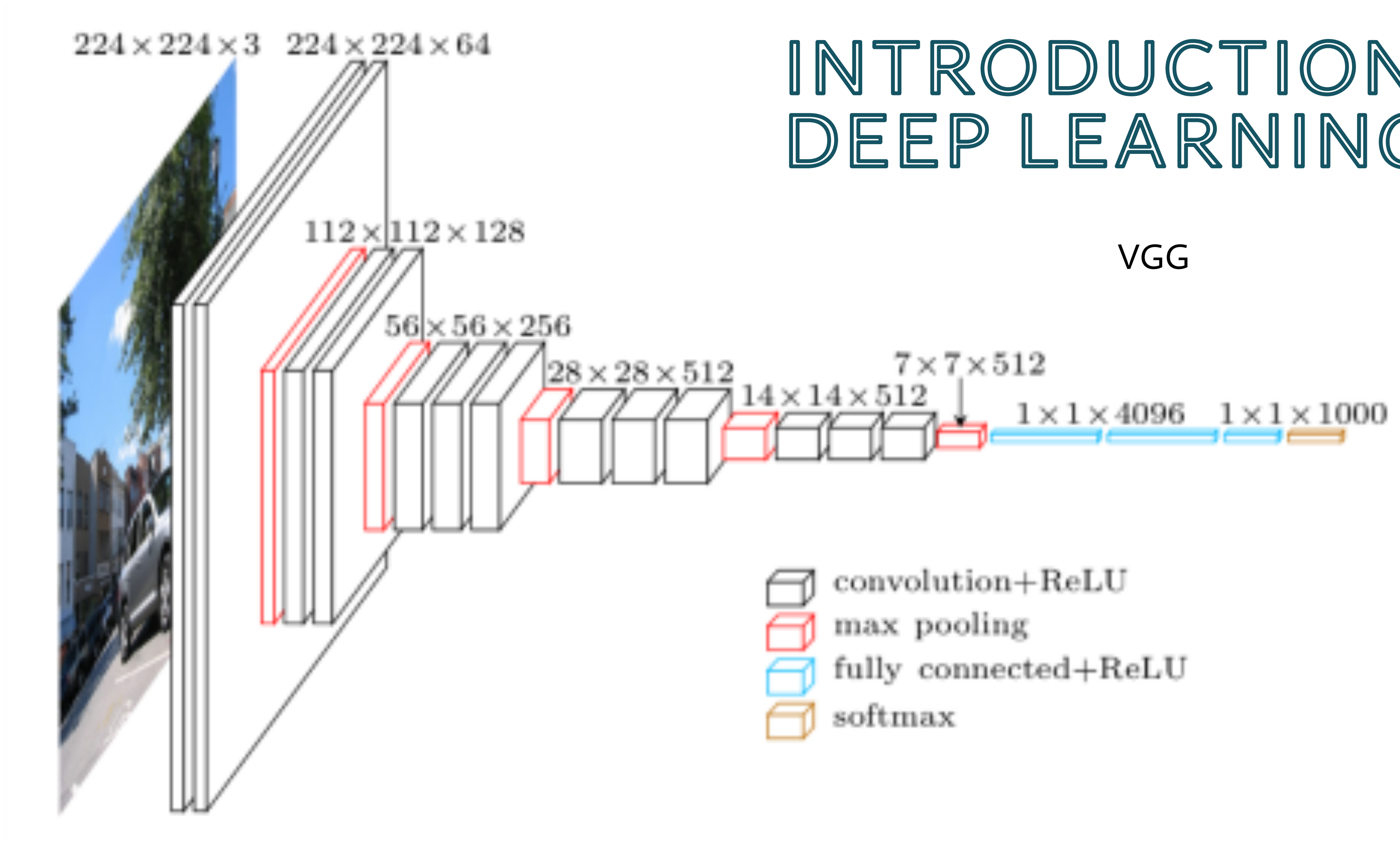
Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

Krizhevsky, A., Sutskever, I., Hinton, G. 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25 (NIPS 2012)

INTRODUCTION TO DEEP LEARNING



VGG

Simonyan, K. and Zisserman, A. 2014. Very Deep Convolutional Neural Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR).
Screen Shot 2565-11-28 at 10.49.24

INTRODUCTION TO DEEP LEARNING

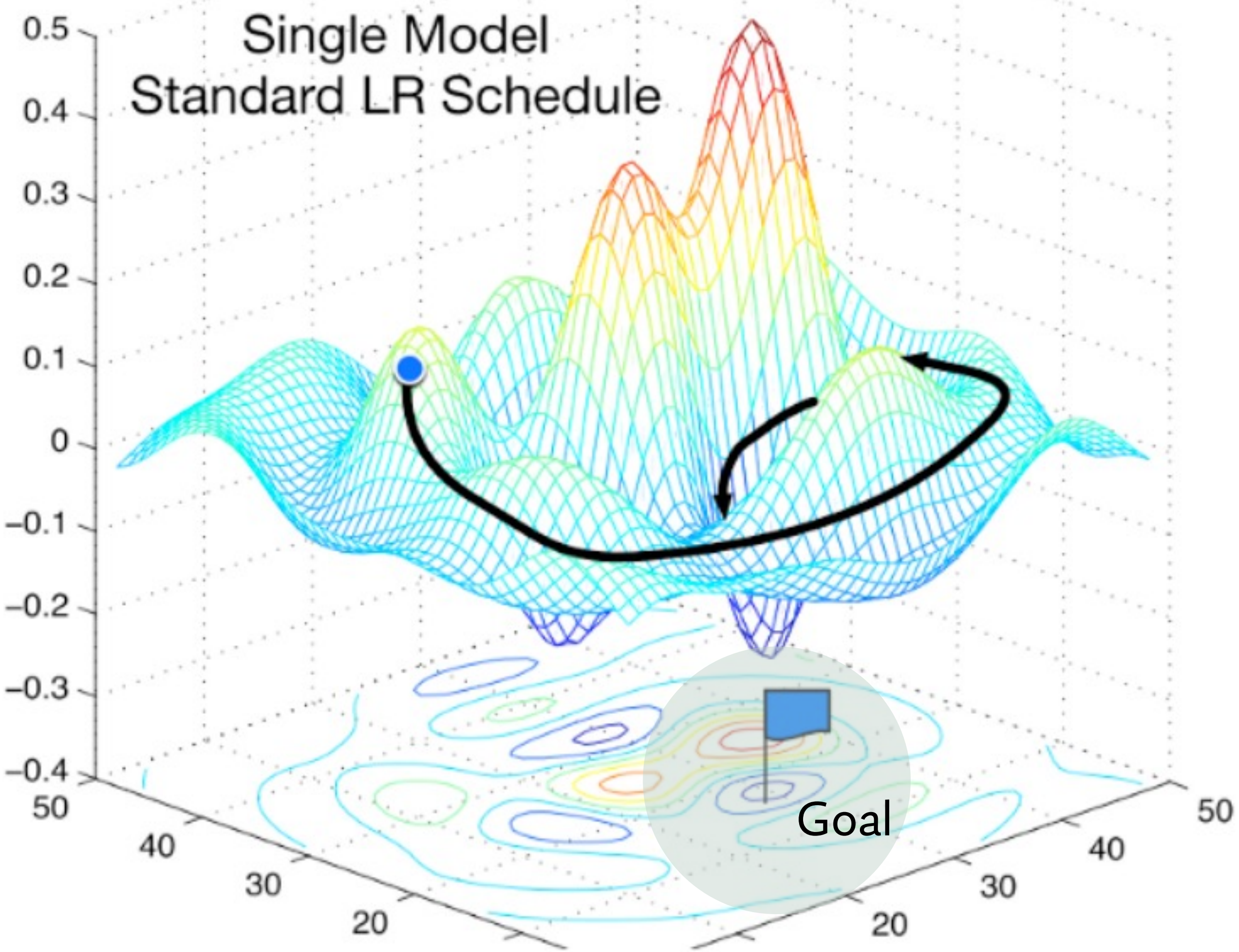
VGG

Table 7: **Comparison with the state of the art in ILSVRC classification**. Our method is denoted as "VGG". Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

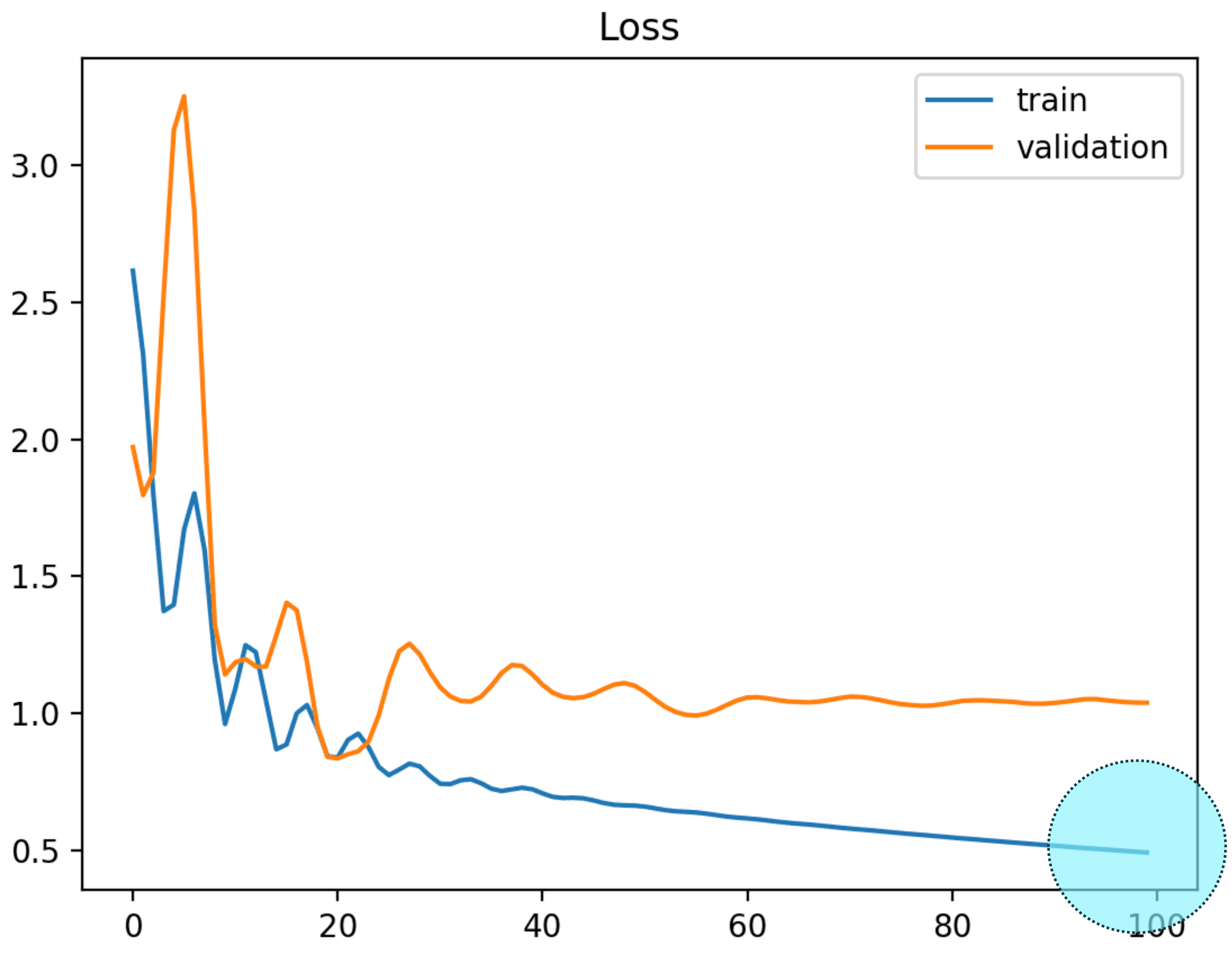
Simonyan, K. and Zisserman, A. 2014. Very Deep Convolutional Neural Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR).
Screen Shot 2565-11-28 at 10.49.24

TRAIN DEEP LEARNING MODEL



- In typical deep learning, the model converges to a <u>minimum</u> at the end of the training.

Huang, et al. (2017). Snapshot Ensembles: Train 1, get M for free. International Conference on Learning Representations



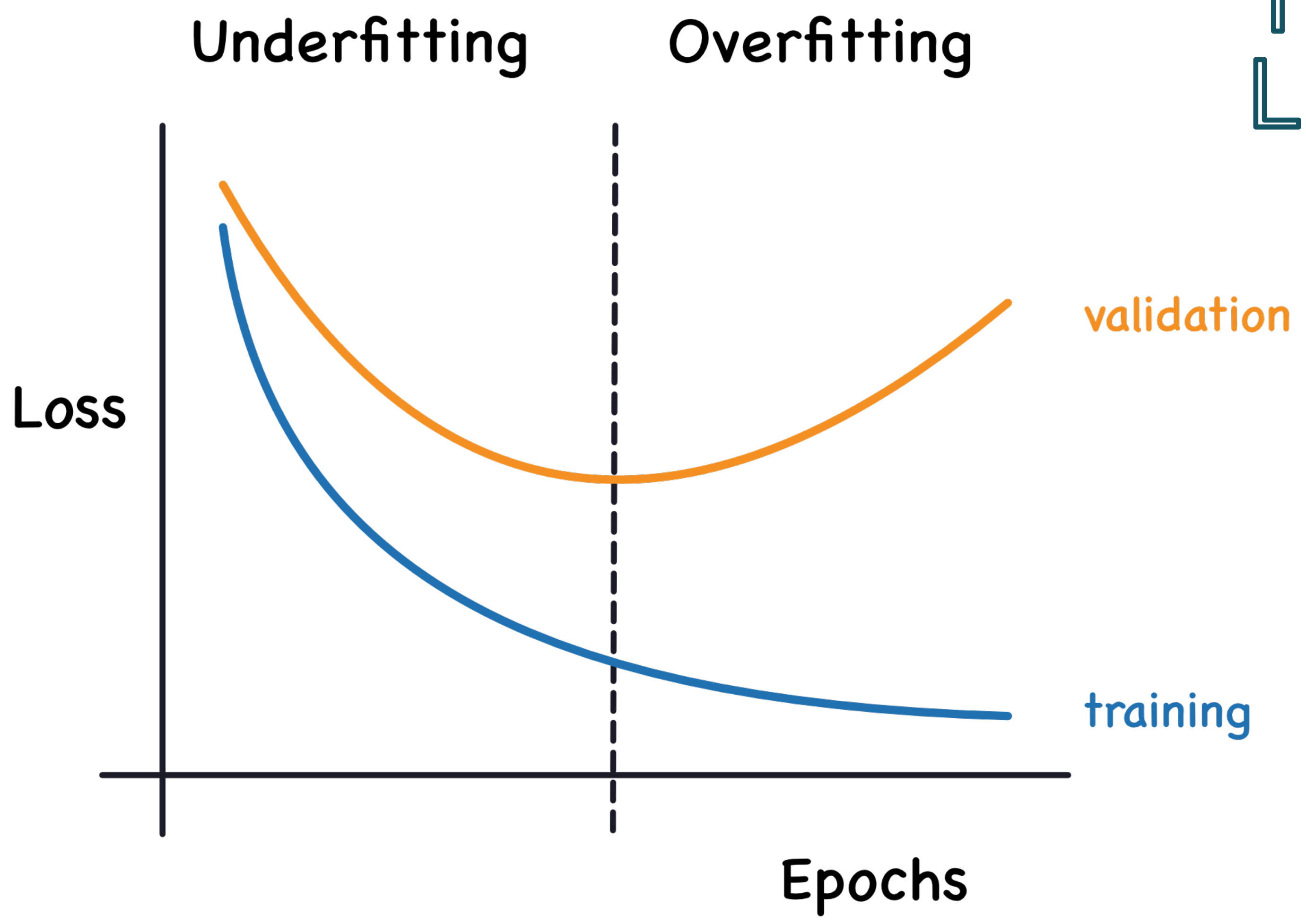
TRAIN DEEP LEARNING MODEL

- The objective of training the deep learning model is to find *minimum training loss/error*.

minimum training loss

https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

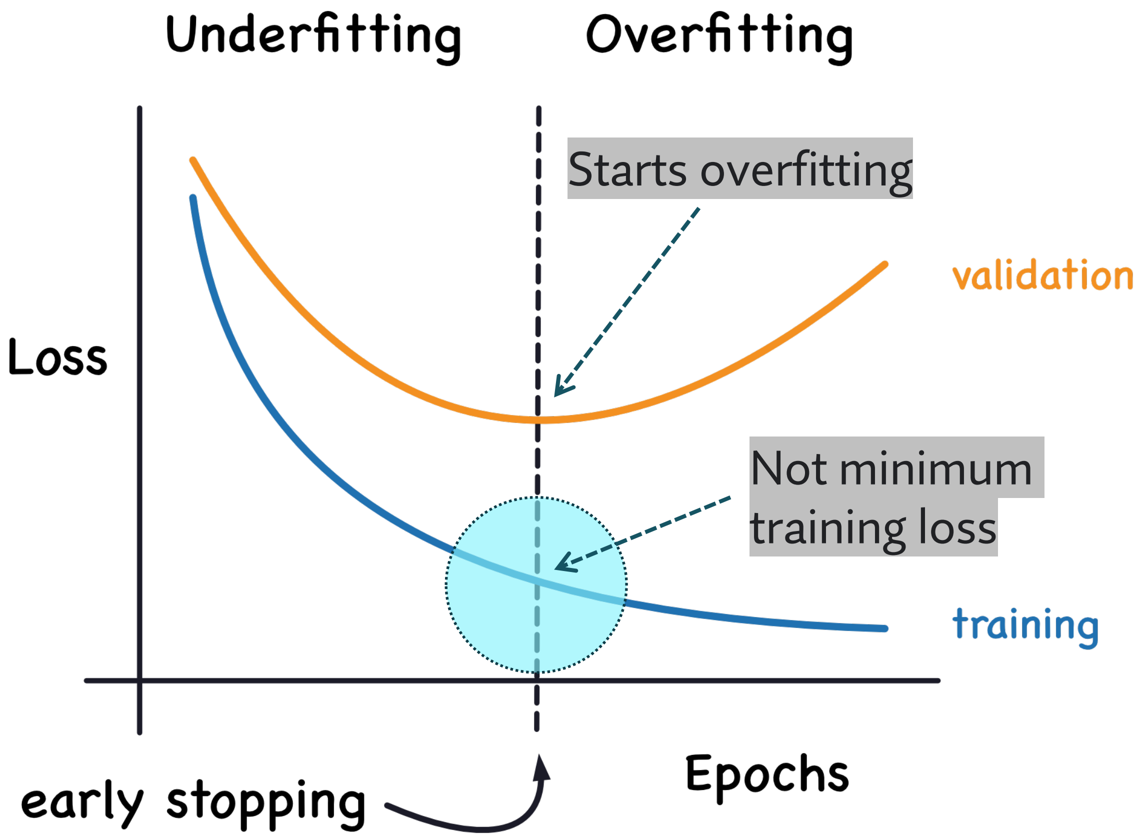
TRAIN DEEP LEARNING MODEL



- **Overfitting problems** appear when the model has a high variance. The model performs well on the training set but does not perform accurately in the <u>validation/test/evaluation/ unseen set</u>.

https://www.kaggle.com/code/ryanholbrook/overfitting-and-underfitting

TRAIN DEEP LEARNING MODEL



- The simple way to avoid the overfitting problem is to stop training the model earlier, called **early stopping.**

https://www.kaggle.com/code/ryanholbrook/overfitting-and-underfitting

TRAIN DEEP LEARNING MODEL

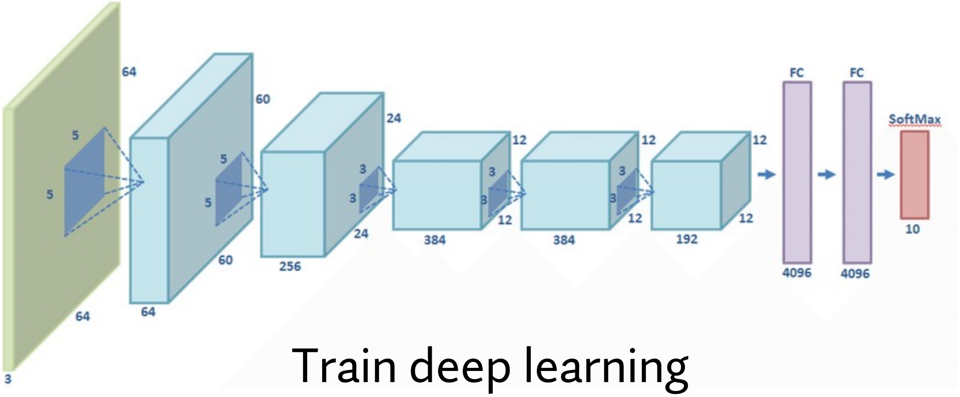
Do we need a ton of training data to train deep learning models?

SCRATCH AND TRANSFER LEARNING

Pawara et al. (2017) trained the deep learning models on the Agriplant dataset, which has 2,400 training images.

Your dataset

Scratch Learning



Train deep learning model

Number of Iterations: 50,000 epochs

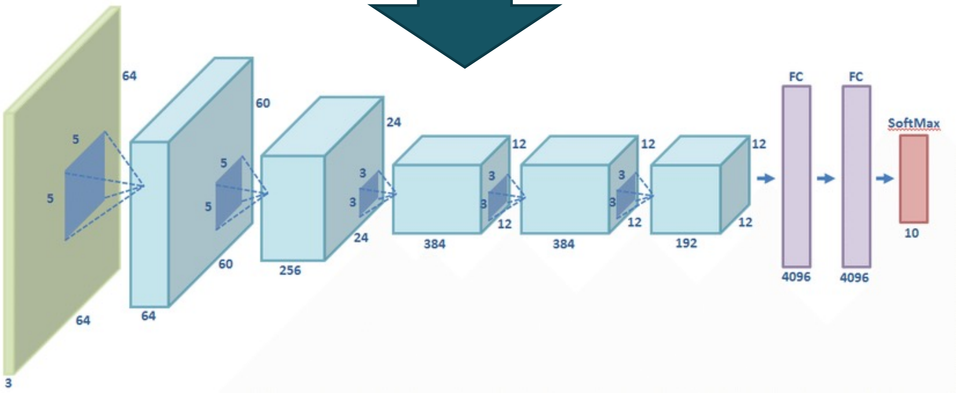
AlexNet and GoogLeNet achieved with an accuracy of 89.53% and 93.33%.

Pawara, P., Okafor, E., Surinta, O., Schomaker, L.R.B. and Wiering, M.A. (2017). Comparing Local Descriptors and Bags of Visual Words to Deep Convolutional Neural Networks for Plant Recognition, in Pattern Recognition Applications and Methods (ICPRAM), The 6th International Conference on, 479-486.

SCRATCH AND TRANSFER LEARNING

ILSVRC. There are roughly **1.2 million training images**, 50,000 validation images, and 150,000 testing images.

1.2 million training images

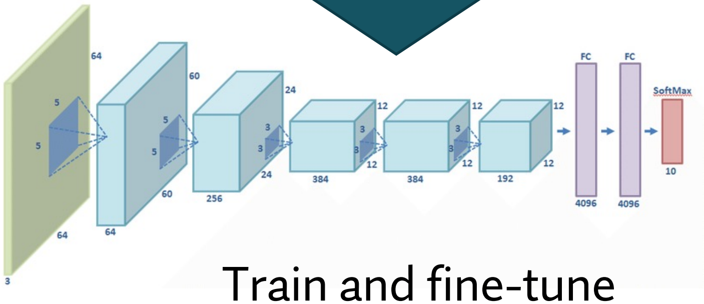


Pretrained model

Transfer Learning

2,400 training images

Your dataset



Train and fine-tune a model

Number of Iterations: 20,000 epochs

AlexNet and GoogLeNet achieved with an accuracy of 96.37% and 98.33%.

SCRATCH AND TRANSFER LEARNING

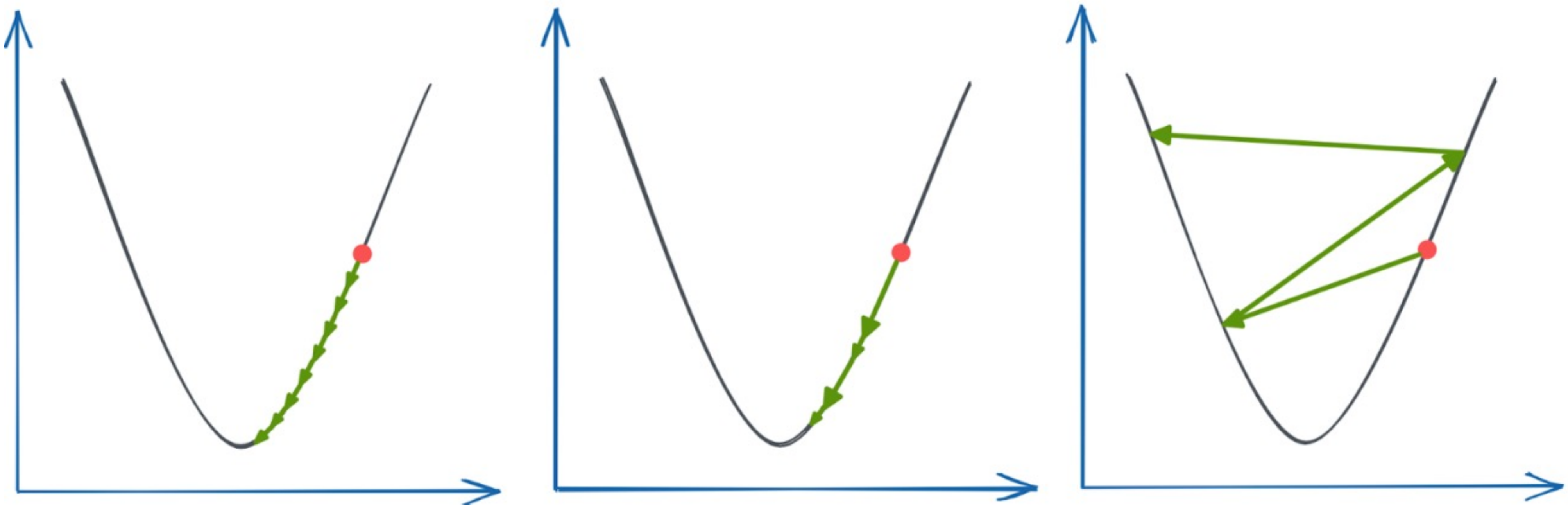
Table 3: Test Accuracy comparison among all techniques on three plant datasets.

Methods	AgrilPlant	LeafSnap	Folio
HOG with KNN	38.13 ± 0.53	58.51 ± 2.47	84.30 ± 1.62
HOG-BOW with MLP	74.63 ± 2.16	79.27 ± 3.36	92.37 ± 1.78
HOG-BOW with SVM	79.43 ± 1.68	72.63 ± 0.38	92.78 ± 2.17
AlexNet scratch	89.53 ± 0.61	76.67 ± 0.56	84.83 ± 2.85
AlexNet fine-tuned	96.37 ± 0.83	89.51 ± 0.75	$\mathbf{97.67 \pm 1.60}$
GoogleNet scratch	93.33 ± 1.24	89.62 ± 0.50	89.75 ± 1.74
GoogleNet fine-tuned	$\mathbf{98.33 \pm 0.51}$	$\mathbf{97.66 \pm 0.34}$	$\mathbf{97.63 \pm 1.84}$

Pawara, P., Okafor, E., **Surinta, O.,** Schomaker, L.R.B. and Wiering, M.A. (2017). Comparing Local Descriptors and Bags of Visual Words to Deep Convolutional Neural Networks for Plant Recognition, in Pattern Recognition Applications and Methods (ICPRAM), The 6th International Conference on, 479-486.

HYPERPARAMETERS TUNING

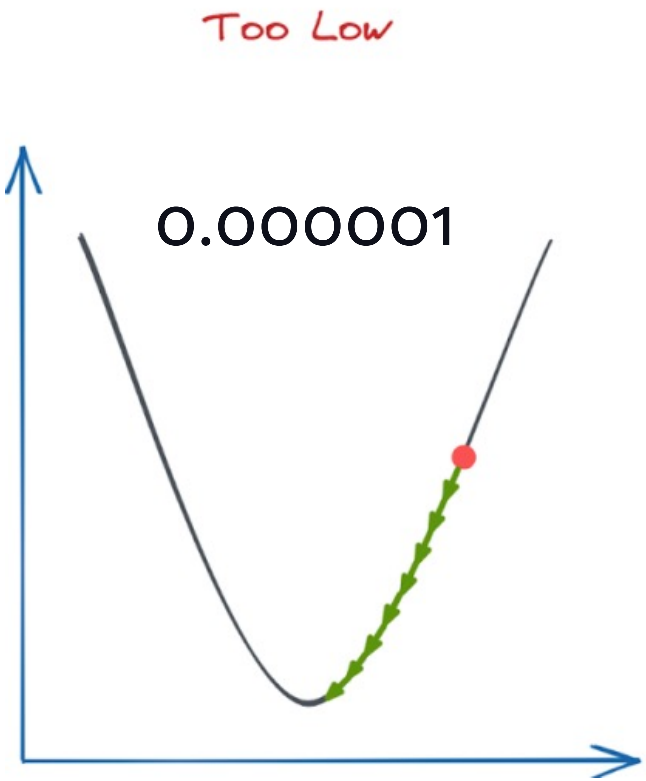
Which graph is the best learning rate tuning?



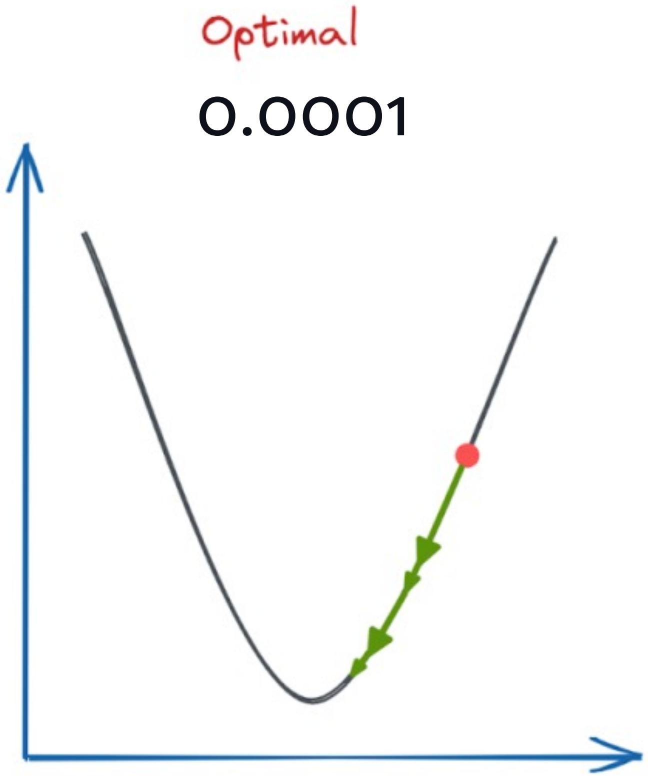
https://analyticsindiamag.com/how-to-automate-finding-the-optimal-learning-rate/

Learning Rate

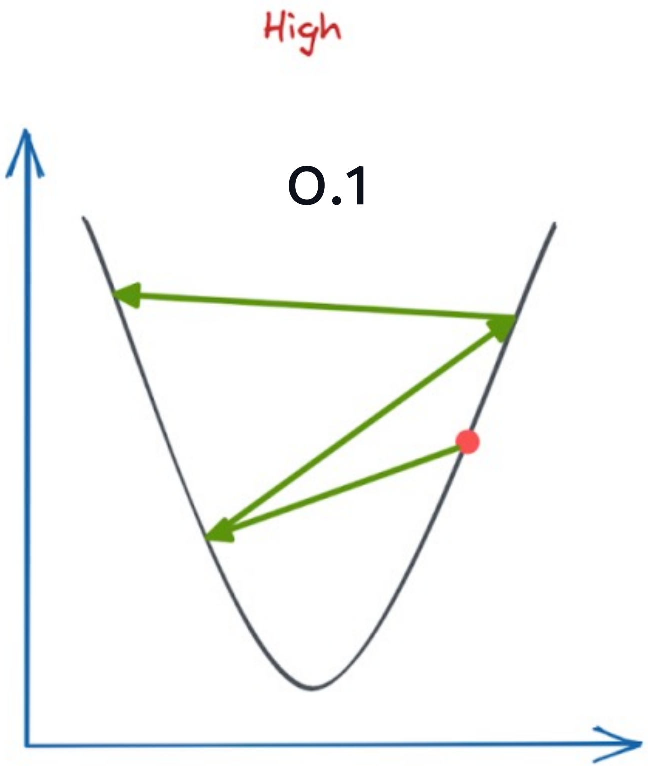
HYPERPARAMETERS TUNING



It takes so much time to converge to a minimum.

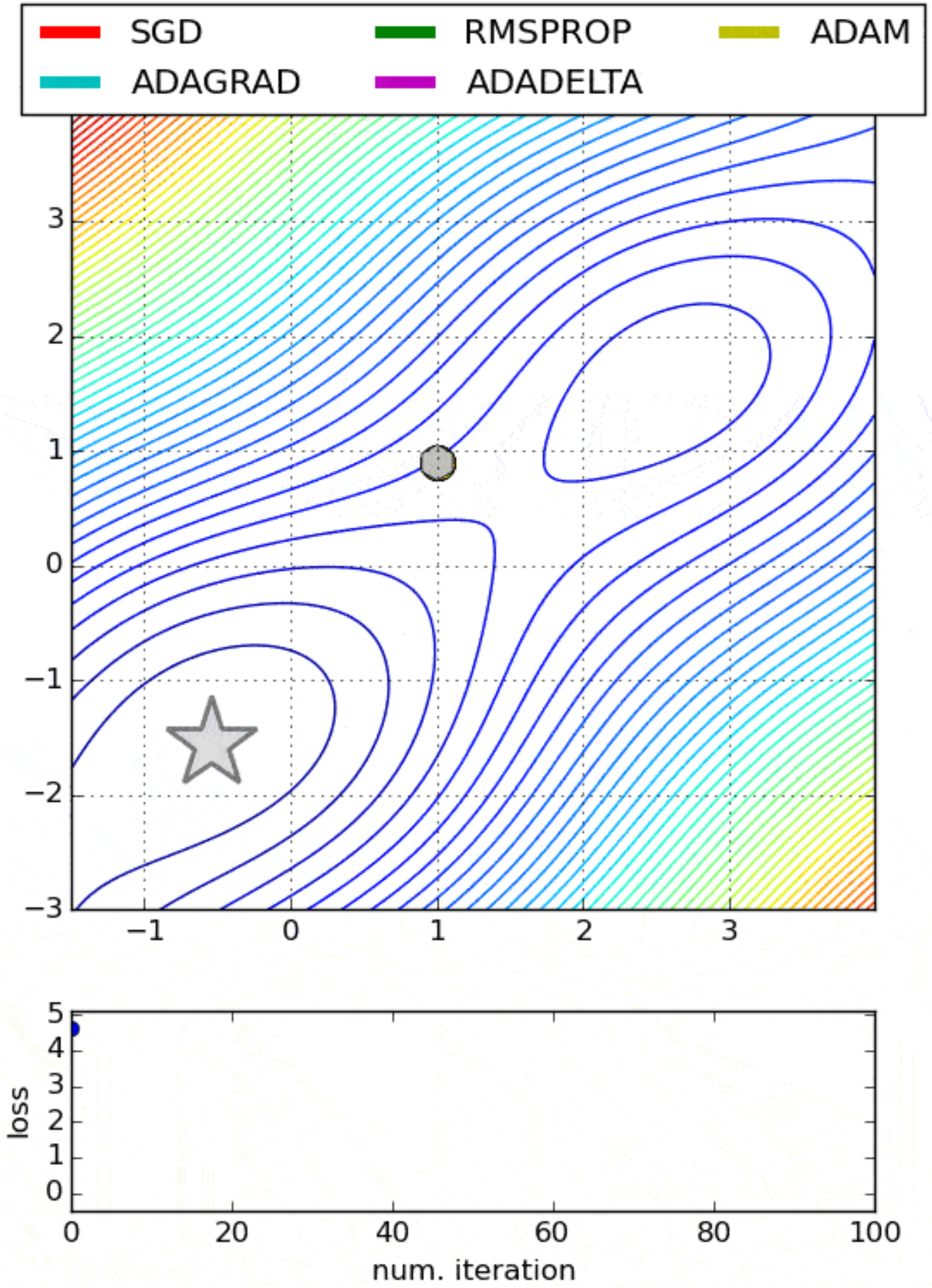


The training loss jumps to a minimum relatively fast.



Sometimes it never converges to a minimum.

https://analyticsindiamag.com/how-to-automate-finding-the-optimal-learning-rate/



https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6

Optimizers

HYPERPARAMETERS TUNING

- Gradient Descent
- Stochastic Gradient Descent (SGD)
- Adagrad
- AdaDelta
- Adam
- RMSProp

Optimizers & Learning rate

HYPERPARAMETERS TUNING

Aerial images

TABLE 1. Performances of the convolutional neural network architectures on the EcoCropsAID dataset

Model	Optimizer	Batch size	Accuracy (%)	Training time	No. of parameters
InceptionResNetV2	SGD	16	48.00	29 mins	54,828,261
MobileNetV2	SGD	32	48.40	5 mins	2,571,589
DenseNet201	SGD	64	50.75	15 mins	18,792,389
Xception	Adam	16	52.99	21 mins	21,885,485
ResNet152V2	SGD	64	59.87	19 mins	58,833,413
NASNetLarge	Adam	8	62.29	1 h 21 mins	87,356,183
VGG19	SGD	64	85.92	12 mins	20,149,829
VGG16	SGD	16	**87.57**	11 mins	14,840,133

Noppitak, S. and Surinta, O. (2021). Ensemble Convolutional Network Architectures for Land Use Classification in Economic Crops Aerial Images. ICIC Express Letters, 15(6), 531-543.

Optimizers & Learning rate

HYPERPARAMETERS TUNING

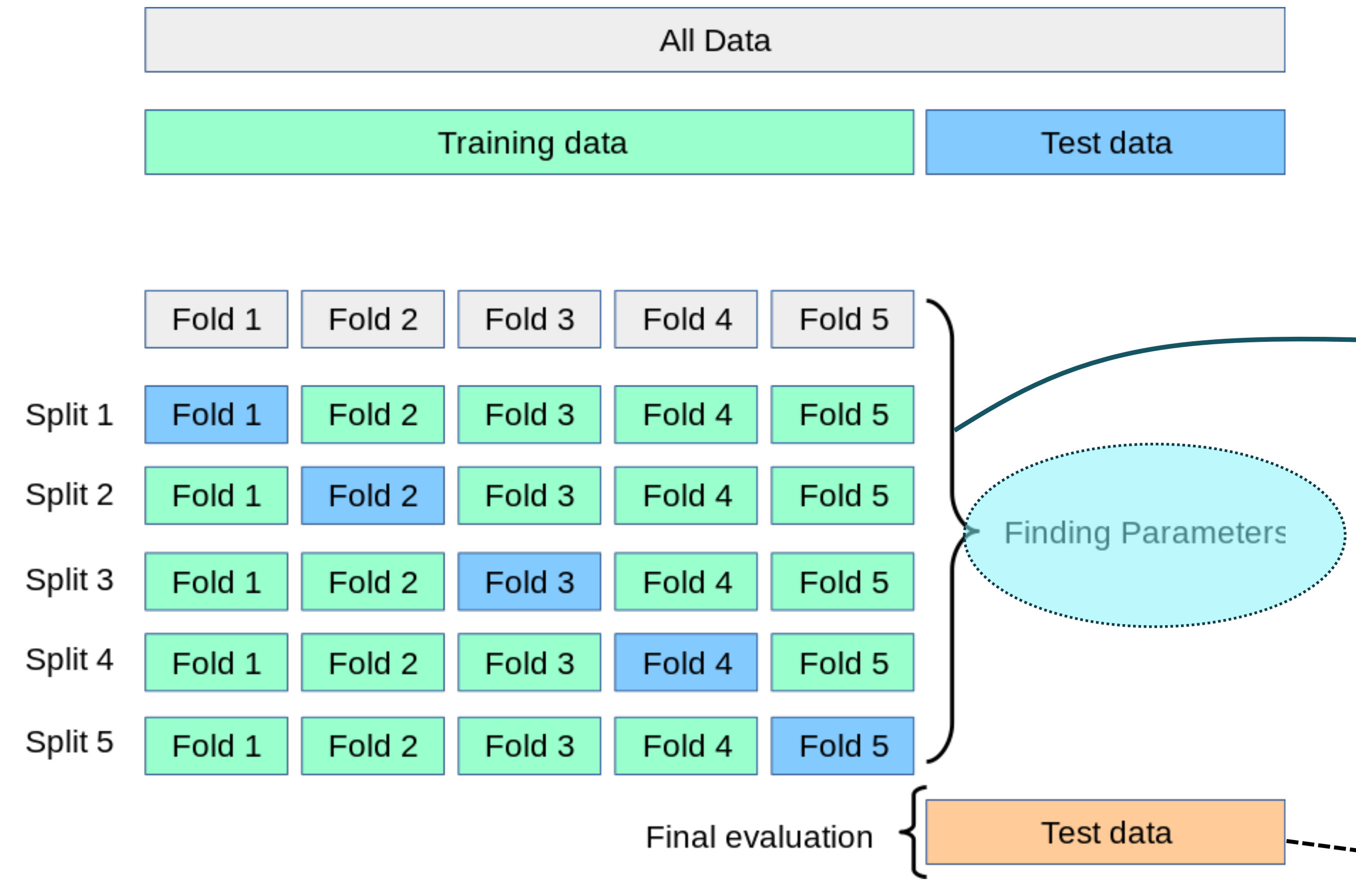
Table 1. The best training hyperparameters and the accuracy (%) of each single model obtained with 5-fold cross-validation and test set on the mulberry leaf dataset

Models	Optimizer	Learning rate	Batch size	Validation (%)	Test accuracy (%)
MobileNetV1	RMSProp	0.0001	8	97.35 ± 0.005	89.83
MobileNetV2	RMSProp	0.0001	16	97.08 ± 0.006	**91.19**
NASNetMobile	RMSProp	0.0001	8	97.38 ± 0.004	86.65
DenseNet121	SGD	0.01	8	$\mathbf{98.61 \pm 0.002}$	90.80
Xception	RMSProp	0.0001	8	97.94 ± 0.006	91.0

Chompookham, T. and Surinta, O. (2021). Ensemble Methods with Deep Convolutional Neural Networks for Plant Leaf Recognition. ICIC Express Letters, 15(6), 553-565.

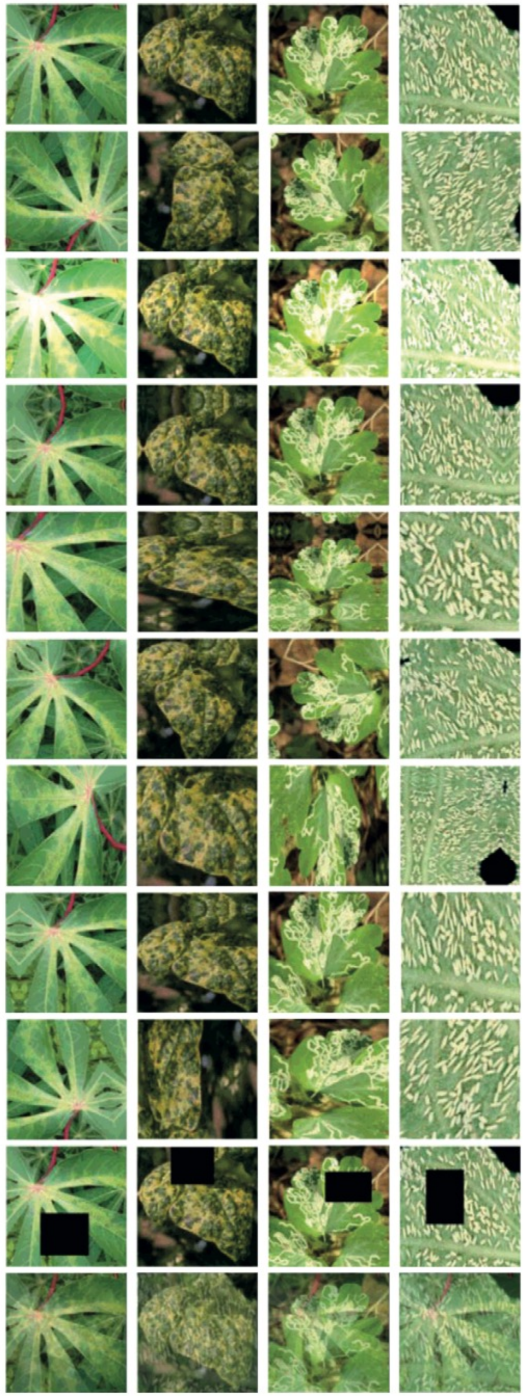
Cross-validation

HYPERPARAMETERS TUNING



Validation (%)	Test accuracy (%)
97.35 ± 0.005	89.83
97.08 ± 0.006	**91.19**
97.38 ± 0.004	86.65
$\mathbf{98.61 \pm 0.002}$	90.80
97.94 ± 0.006	91.0

https://scikit-learn.org/stable/modules/cross_validation.html



DATA AUGMENTATION

Examples of the (a) leaf disease images and samples of data augmentation images using (b) rotation, brightness, (d) sift, (e) zoom, (f) rotation+shift, (g) rotation+zoom, (h) shift+zoom, (i) rotation+shift+zoom, (j) cutout, and (k) mixup techniques.

Table 4: Data augmentation techniques and parameters used in the experiment

Data Augmentation Techniques	Parameters
Rotation	[–170, 170]
Brightness	[1, 5]
Width shift	[–0.2, +0.2]
Height shift	[–0.2, +0.2]
Zoom	[0.5, 1.5]
Fill mode	Reflect
Cutout	0.5
Mixup	0.4

Enkvetchakul, P. and Surinta, O. (2022). Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition. Applied Science and Engineering Progress, 15(3), 3810.

DATA AUGMENTATION

Combination

Table 5: MobileNetV2 and NASNetMobile architectures on the leaf disease dataset using different data augmentation techniques

Data Augmentation Methods	MobileNetV2			NASNetMobile		
	Time	Scratch	Fine-Tuning	Time	Scratch	Fine-Tuning
Original image	2 h 12 m	63.08	93.08	4 h 50 m	68.08	92.31
Brightness	20 h 15 m	65.39	90.77	1 d 11 h 30 m	66.92	89.23
Shift		74.62	90.77		75.39	93.08
Rotation		77.69	94.62		83.08	93.85
Zoom		77.69	95.39		64.62	93.01
Shift + Zoom		82.31	93.08		84.62	92.31
Rotation + Zoom		79.23	93.85		76.92	93.08
Rotation + Shift		79.23	95.39		77.69	96.15
Rotation + Shift + Zoom		77.69	90.77		81.54	95.39
Cutout		64.06	93.75		77.34	93.75
Mixup		61.71	89.84		67.18	92.18

Enkvetchakul, P. and Surinta, O. (2022). Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition. Applied Science and Engineering Progress, 15(3), 3810.

DATA AUGMENTATION

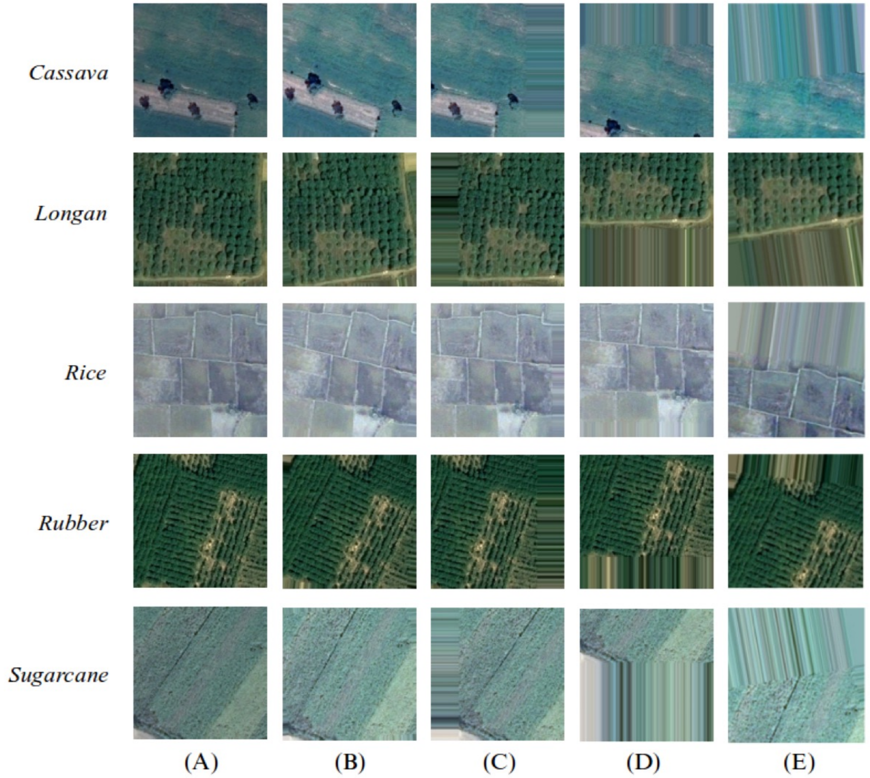


FIGURE 3. Example of data augmentation techniques: (A) Original image, (B) rotation, (C) width shift, (D) height shift, and (E) combination between rotation and width shift

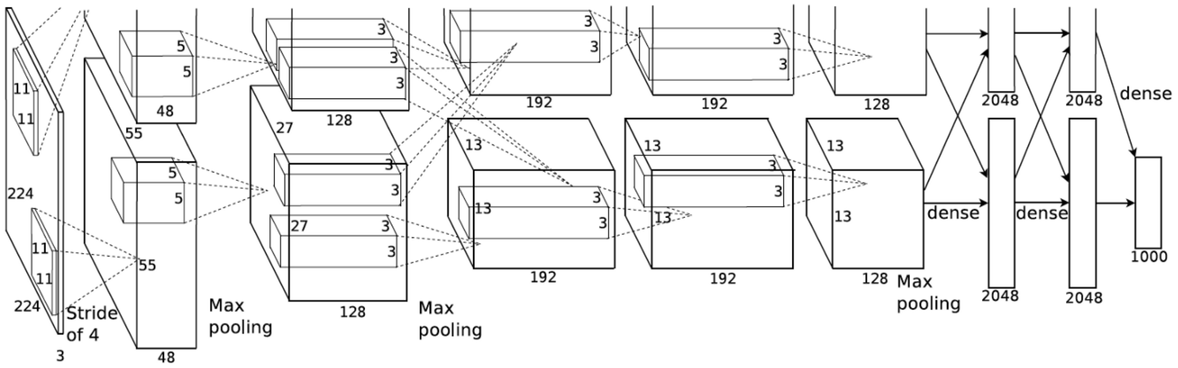
TABLE 2. Test accuracy (%) of the CNN architectures and data augmentation techniques on the EcoCropsAID dataset

Model (Data augmentation)	Optimizer	Batch size	Accuracy	Training time	No. of parameters
NASNetLarge (Hshift)	Adam	8	59.40	1 d 1 h 12 mins	87,356,183
NASNetLarge (Rotation + Hshift)	Adam	8	79.60	1 d 1 h 29 mins	87,356,183
VGG19 (Rotation + Hshift)	SGD	64	86.50	6 h 23 mins	20,149,829
VGG19 (Hshift)	SGD	64	88.30	6 h 20 mins	20,149,829
VGG16 (Rotation + Hshift)	SGD	16	**91.50**	6 h 39 mins	14,840,133
VGG16 (Hshift)	SGD	16	**91.50**	6 h 39 mins	14,840,133

Noppitak, S. and Surinta, O. (2021). Ensemble Convolutional Network Architectures for Land Use Classification in Economic Crops Aerial Images. ICIC Express Letters, 15(6), 531-543.

ENSEMBLE LEARNING

AlexNet



Average outputs from 7CNNs

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

ENSEMBLE LEARNING

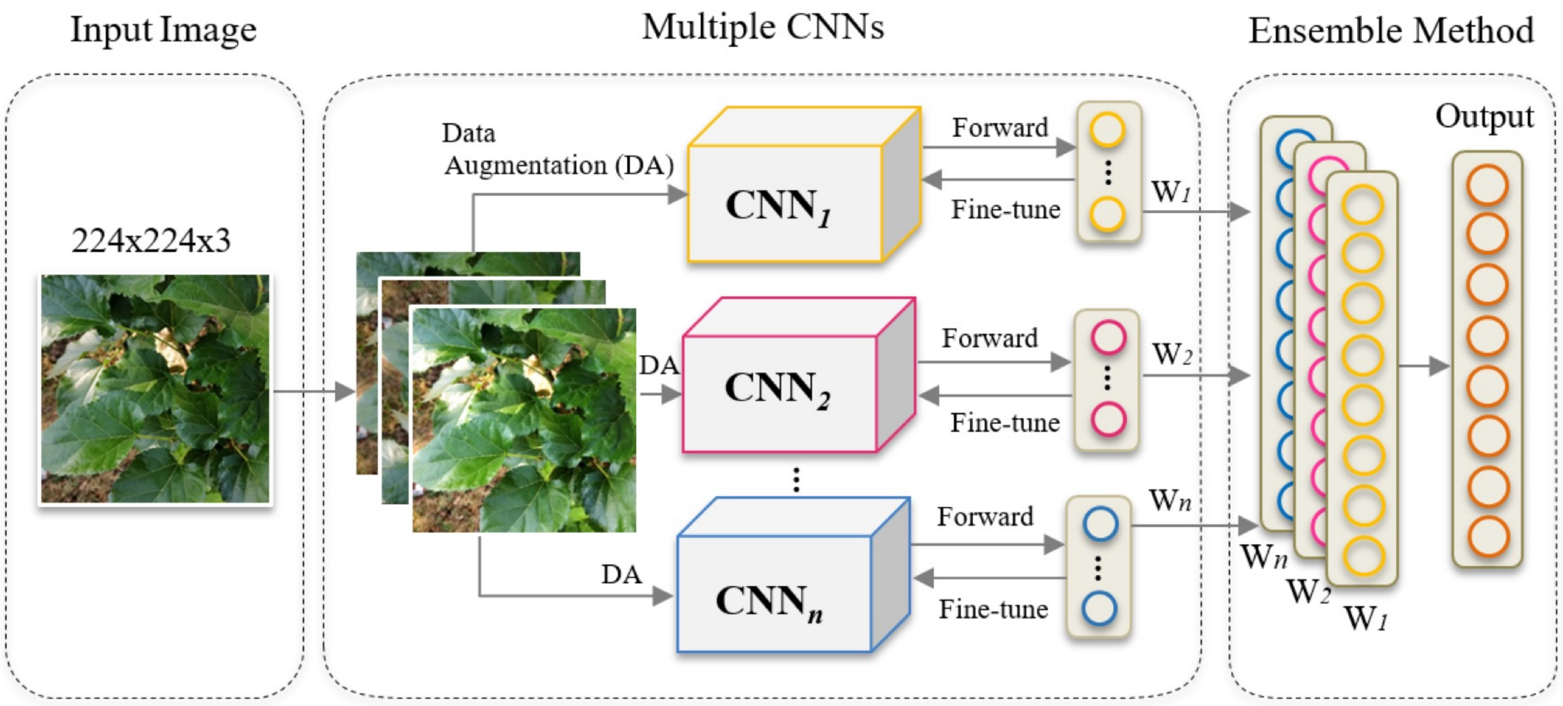


FIGURE 1. The framework of the proposed ensemble CNNs

- Majority vote
- Unweighted average
- Weighted average

Chompookham, T. and Surinta, O. (2021). Ensemble Methods with Deep Convolutional Neural Networks for Plant Leaf Recognition. ICIC Express Letters, 15(6), 553-565.

ENSEMBLE LEARNING

Noppitak and Surinta (2021)

TABLE 3. Performances of the ensemble CNN methods on the EcoCrop-sAID dataset

Model	Ensemble method		
	Unweighted majority vote	Unweighted average	Weighted average
E1	92.00	92.60	92.60
E2	91.90	92.40	92.70
E3	92.30	92.30	92.70
E4	92.50	92.60	**92.80**

Chompookham and Surinta. (2021)

TABLE 5. Performance of the ensemble CNN methods applied on plant leaf datasets

Datasets/DA	Test accuracy (%)					
	Unweighted majority vote		Unweighted average		Weighted average	
	3-EnsCNNs	5-EnsCNNs	3-EnsCNNs	5-EnsCNNs	3-EnsCNNs	5-EnsCNNs
Mulberry leaf dataset						
No DA	92.81	93.65	94.55	94.68	94.49	**94.75**
DA	92.61	92.81	94.03	94.23	94.41	94.55
Tomato leaf dataset						
No DA	99.20	99.20	99.79	99.86	99.86	99.79
DA	99.26	99.20	99.86	99.79	**99.93**	99.86
Corn leaf dataset						
No DA	98.44	98.70	99.45	99.21	**99.47**	99.24
DA	98.44	98.70	99.21	99.21	99.31	99.30

FUSION CNN

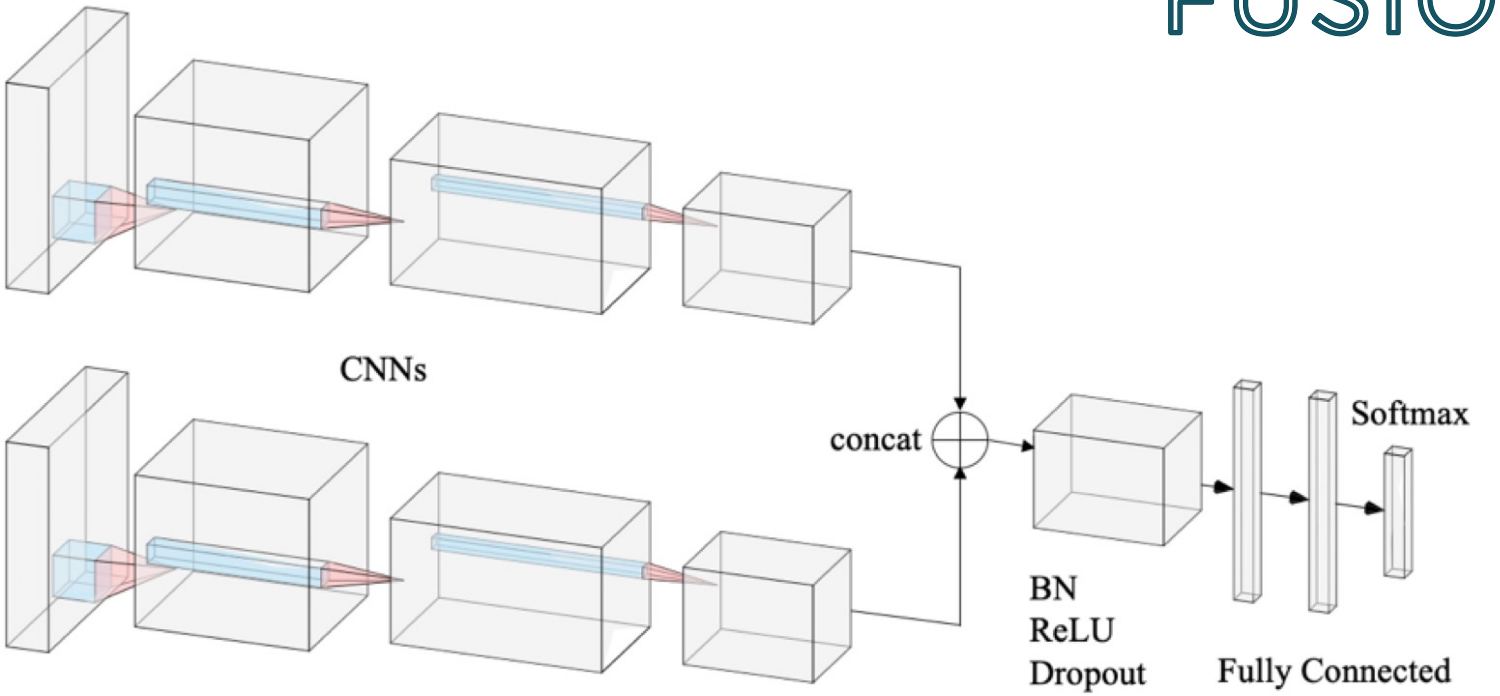
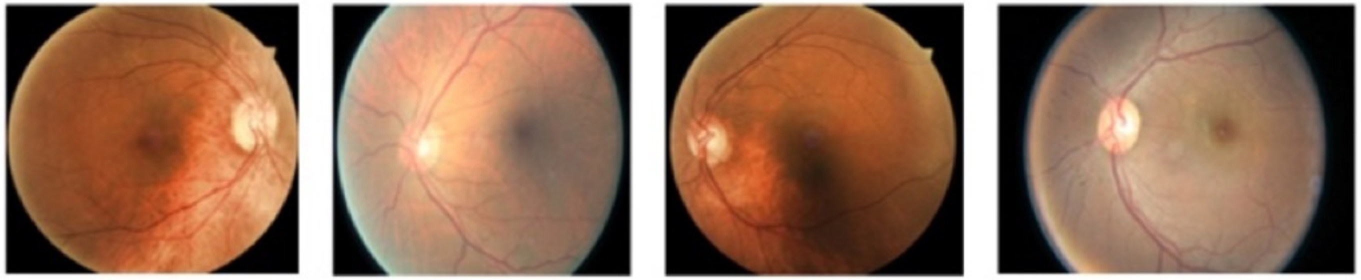


FIGURE 1. Illustration of the fusion CNN architecture

Saichua, P. and Surinta, O. (2022). Comparative Study between Ensemble and Fusion Convolutional Neural Networks for Diabetic Retinopathy Classification. ICIC Express Letters, 16(4), 401-408.



FUSION CNN

TABLE 3. Performance evaluation of the fusion CNNs

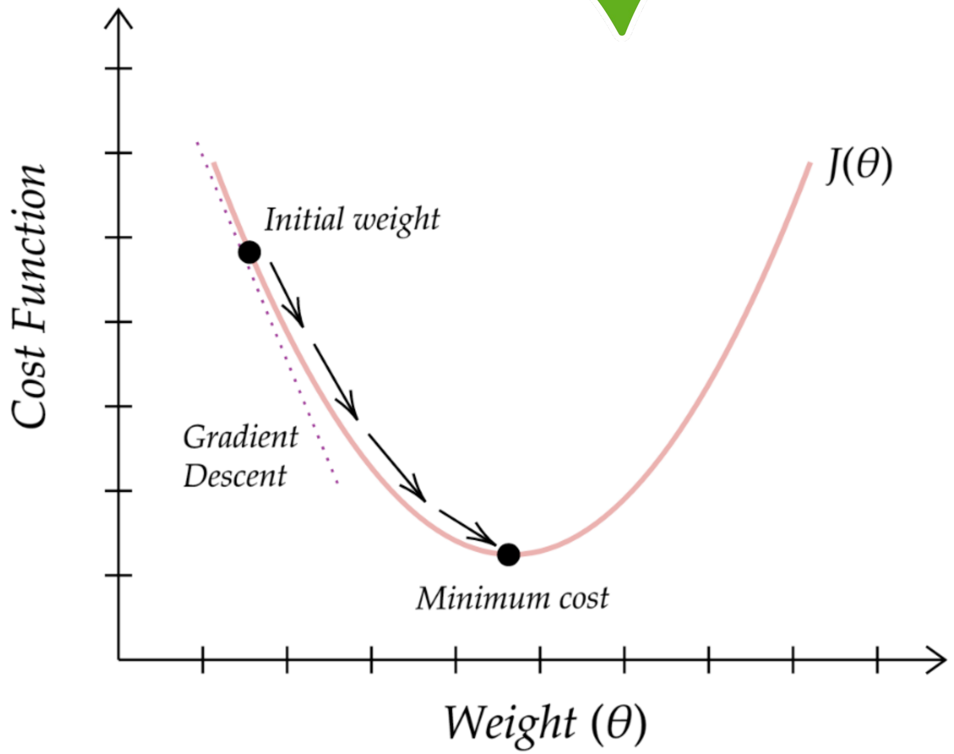
Fusion CNNs	Dense sizes	No. of dense layers	Dropout	Training time	Test Accuracy (%)	Test Loss
Xception+InceptionV3	1024	2	0.4	53 min	**86.30**	**0.33**
Xception+DenseNet121	2048	2	0.2	33 min	85.45	0.34
Xception+ResNet50V2	4096	1	No	30 min	85.11	0.36
Xception+ResNet50	1024	1	0.1	32 min	85.07	0.4

TABLE 4. Performance evaluation of the ensemble CNNs

Ensemble CNNs	Accuracy (%) of ensemble learning methods		
	Unweighted average method	Weighted average method	Weight parameters
Xception+InceptionV3	85.92	**86.11**	0.6, 0.4
ResNet50+DenseNet121	84.30	84.81	0.3, 0.7
ResNet50V2+DenseNet121	84.22	84.39	0.3, 0.7
ResNet50+ResNet50V2	83.65	83.73	0.3, 0.7

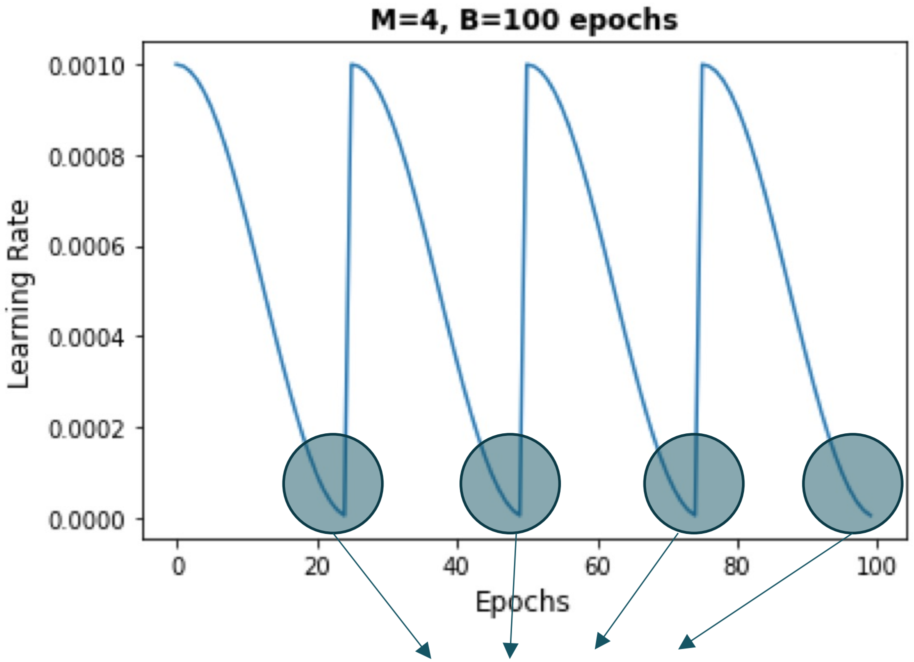
SNAPSHOT ENSEMBLE CNN

Ensemble methods	Accuracy (%)
Unweighted ensemble	68.54
Weighted ensemble	69.32
Snapshot ensemble ($M = 4$)	**69.51**



Learning rate schedule

$$\eta = \frac{\eta_{init}}{2}\left(\cos\left(\frac{\pi\, mod(t-1, [T/M])}{[T/M]}\right) + 1\right)$$

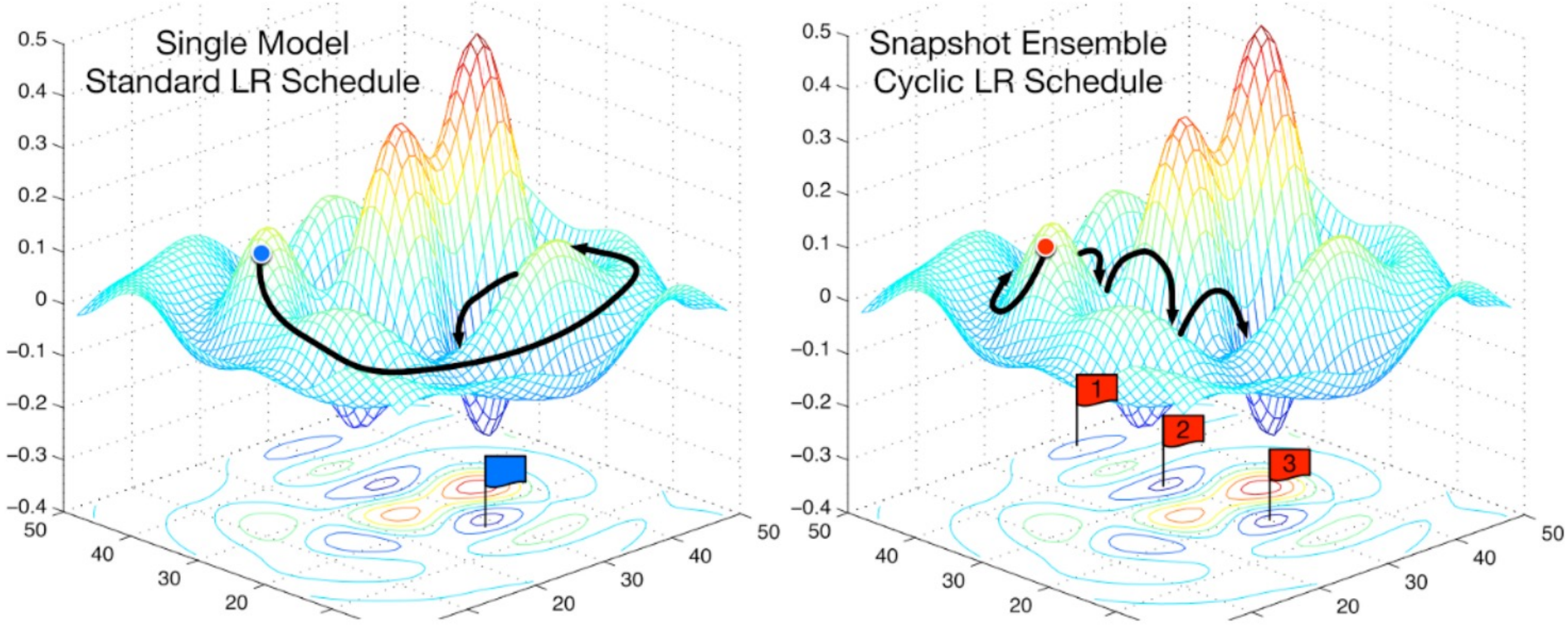


Snapshot

Ensemble

Puangsuwan, T. and Surinta, O. (2021). Enhancement of Plant Disease Classification Based on Snapshot Ensemble Convolutional Neural Network. ICIC Express Letters, 15(6), 669-680.

SNAPSHOT ENSEMBLE CNN



Huang, et al. (2017). Snapshot Ensembles: Train 1, get M for free. International Conference on Learning Representations

SNAPSHOT ENSEMBLE CNN

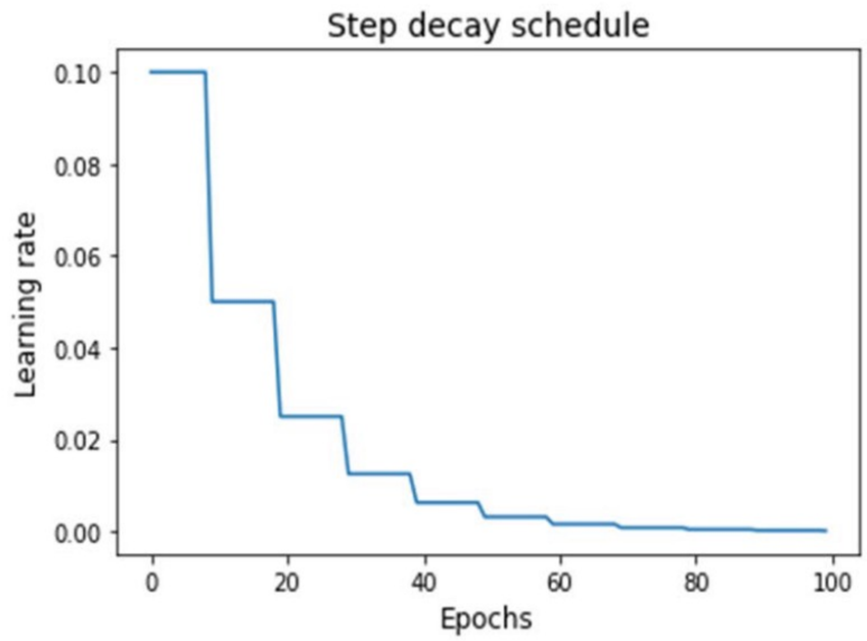


FIGURE 2. **Illustration of the step decay schedule.**

Drop Cyclic Cosine Learning Rate Schedule

$$\eta = \frac{\eta_{init}}{2} \times drop^{floor((1+t)/c)\times(1+cos(\frac{\pi mod([t,b],[T,/Mb])}{[T/Mb]}))}$$

Noppitak, S. and Surinta, O. (2022). dropCyclic: Snapshot Ensemble Convolutional Neural Network Based on a New Learning Rate Schedule for Land Use Classification. IEEE Access, 10, 60725-60737.

SNAPSHOT ENSEMBLE CNN

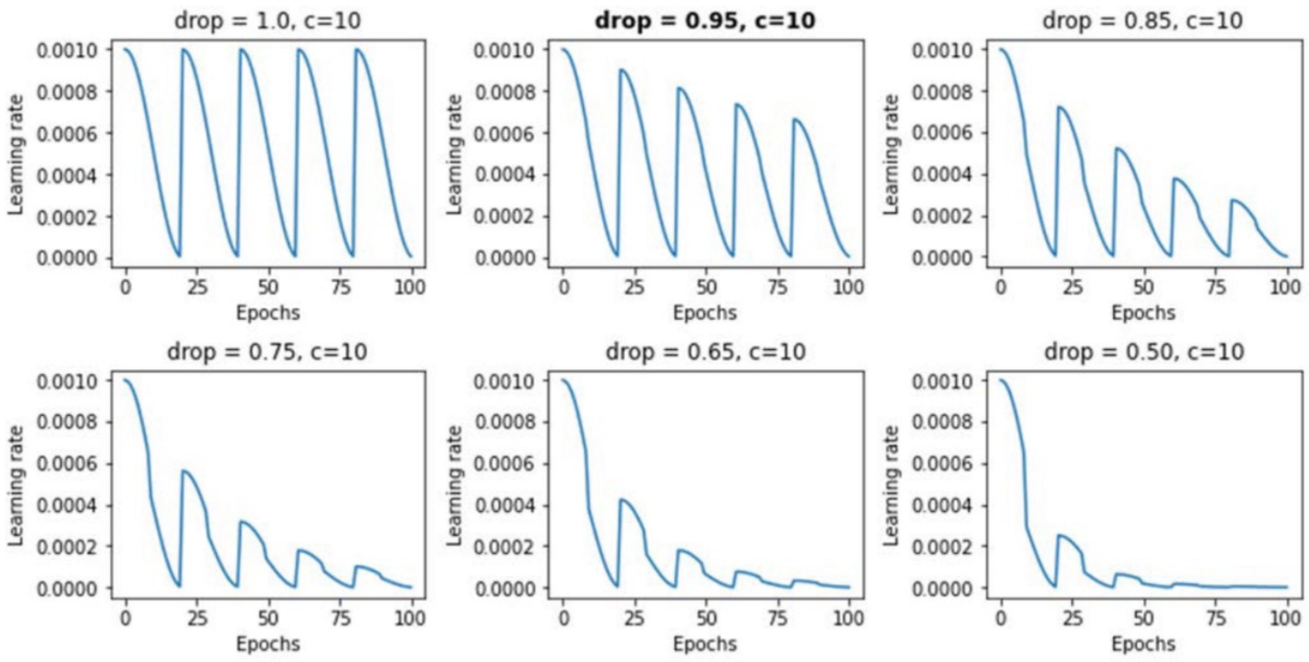


TABLE 3. **Classification performances (LD, mean validation accuracy, standard deviation and test accuracy) of the snapshot ensemble CNN using different learning rate schedules: CCA CCA, MMCCLP, and dropCyclic and training with different state-of-the-art CNNs: MobileNetV2, VGG16, VGG19, on the UCM dataset.**

CNNs	LR methods	LD	Validation	Test
MobileNetV2	CCA	**0.0422**	97.30±0.0085	97.14
	MMCCLR	0.0626	97.10±0.0081	96.43
	dropCyclic	0.0560	**97.38±0.0050**	**97.38**
VGG16	CCA	0.1650	95.47±0.0050	**93.57**
	MMCCLR	**0.1371**	95.83±0.0042	93.10
	dropCyclic	0.1965	**96.51±0.0054**	92.62
VGG19	CCA	0.2285	**96.98±0.0097**	93.10
	MMCCLR	0.1475	96.03±0.0056	93.57
	dropCyclic	**0.0896**	96.63±0.0020	**94.76**

CONCLUSION

- There are many methods to improve the performance of deep learning models, such as tuning the hyperparameters (learning rate, optimization algorithms), increasing the number of training data, generating new data based on existing data, etc.

- However, the deep learning model will face the overfitting problem that it always obtains high accuracy performance and worsens when used in the real world.

48th STT

SCIENCE AND TECHNOLOGY
FOR ADVANCING TOWARDS SDGs

THANK YOU FOR YOUR ATTENTION