From Pixels to Predictions: Building a Deep Learning Model

MARTYNA ANTAS

Introduction

THE CHALLENGE: BUILDING A MACHINE LEARNING MODEL TO INTERPRET AND CLASSIFY VISUAL DATA.

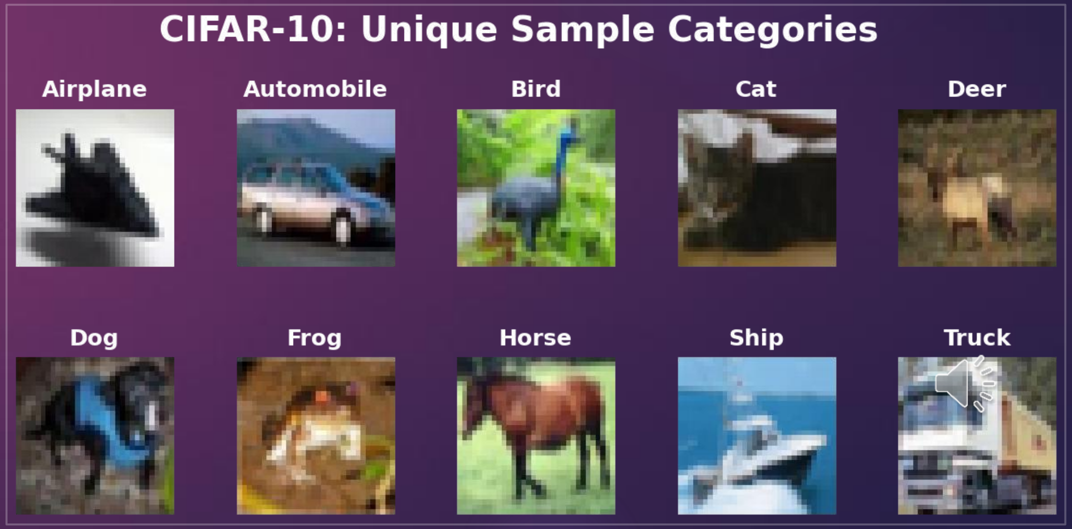
THE PROCESS: DESIGNING AND FINE-TUNING NEURAL NETWORKS TO BALANCE SIMPLICITY AND PERFORMANCE.

THE JOURNEY: TACKLING TECHNICAL CHALLENGES AND UNCOVERING INSIGHTS INTO THE POWER AND LIMITATIONS OF NEURAL NETWORKS.

(LeCun, Bengio, and Hinton, 2015)

CIFAR-10

- **60,000 images spanning 10 categories** : 50,000 allocated for training purposes and 10,000 reserved for testing.
- **Challenge :** Identifying patterns in images sized at 32x32 pixels.
- **Complexity :** Distinguishing between visually similar classes presents a significant challenge.
- **Versatility :** Perfect for feature extraction, augmentation, and the evaluation of models.



(Krizhevsky and Hinton, 2009, pp. 5–6)

Framework

Framework	Advantages	Disadvantages
Keras	Simple, beginner-friendly, fast prototyping	Limited flexibility for complex use cases
PyTorch	Flexible, dynamic computation graph, great for research	Steeper learning curve, less deployment focus
TensorFlow + Keras	Combines simplicity with scalability, production-ready tools	More complex than Keras alone, steeper learning curve

(Medavarapu, 2024; Dai et al., 2022; Wang et al., 2024)

Network Architecture

- **First Layer:** Extracts basic visual features (edges, textures).
- **Second Layer:** Captures more detailed patterns, such as shapes and contours.

```
# First convolutional layer
model.add(Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(32, 32, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))

# Second convolutional layer
model.add(Conv2D(64, (3, 3), activation='relu', padding='same', kernel_constraint=MaxNorm(3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

(Srivastava et al., 2014; Brownlee, 2019)

Network Architecture

- Third Layer: Focuses on high-level abstractions and class differentiation.
- Dense Layer: Combines features for final decisions, followed by a softmax output for 10-class probabilities.

```
# Third convolutional layer
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.3))

# Flatten the output for the dense layers
model.add(Flatten())

# Fully connected dense layer
model.add(Dense(256, activation='relu', kernel_constraint=MaxNorm(3)))
model.add(Dropout(0.5))

# Output layer
model.add(Dense(10, activation='softmax'))
```

(Krizhevsky, Sutskever, and Hinton, 2012; Srivastava et al., 2014; Murugan, 2018)

Activation and Loss Functions

ReLU:

- Tackles the vanishing gradient problem.
- Outputs zero for negatives, scales positives linearly.

Softmax:

- Converts outputs into probabilities.
- Perfect for multi-class classification.

Categorical Cross-Entropy:

- Minimises prediction errors.

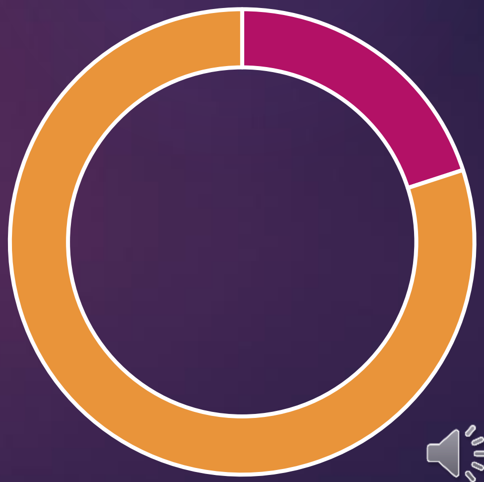
(Asadi and Jiang, 2020; Franke and Degen, 2023; Mao, Mohri, and Zhong, 2023)

Data Partitioning

- **80-20 Split**: Balanced, efficient, and dependable
- **90-10 Split**: Tested but caused instability and overfitting.
- **K-Fold Cross-Validation**: Improved generalisation but required excessive resources.

Final Decision: 80-20 split for its stability and practicality.

Data Partitioning



(Hebart and Vicente, 2016; Xu and Goodacre, 2018)

Preprocessing

Normalisation: Scaled pixel values to 0.0–1.0 for consistent input and faster convergence.

(Huang et al., 2020)

One-Hot Encoding: Converted class labels for compatibility with categorical cross-entropy.

(Mao, Mohri, and Zhong, 2023)

Data Augmentation: Added random rotations, shifts, and flips to enhance generalisation.

(Shorten and Khoshgoftaar, 2019)

Training Process

Initial Setup: 25 epochs, learning rate of 0.1, batch size of 32.

First Adjustment: Increased to 50 epochs, but early stopping triggered at 30. Adjusted learning rate to 0.01 with decay.

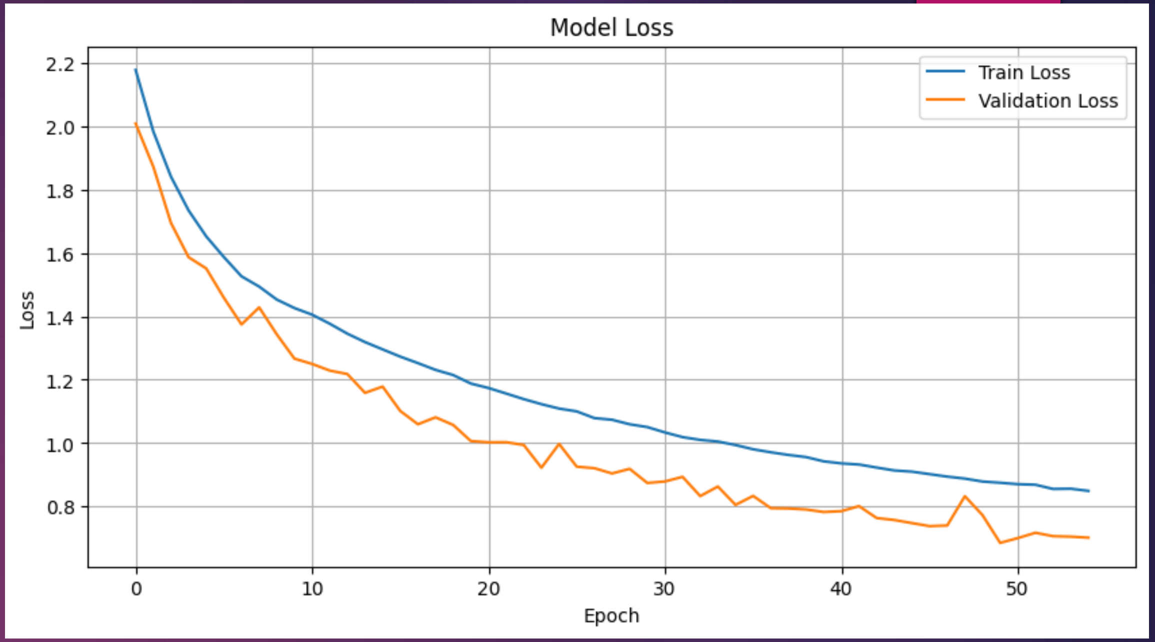
Second Adjustment: Extended to 75 epochs; early stopping occurred at 55 and later at 66 epochs.

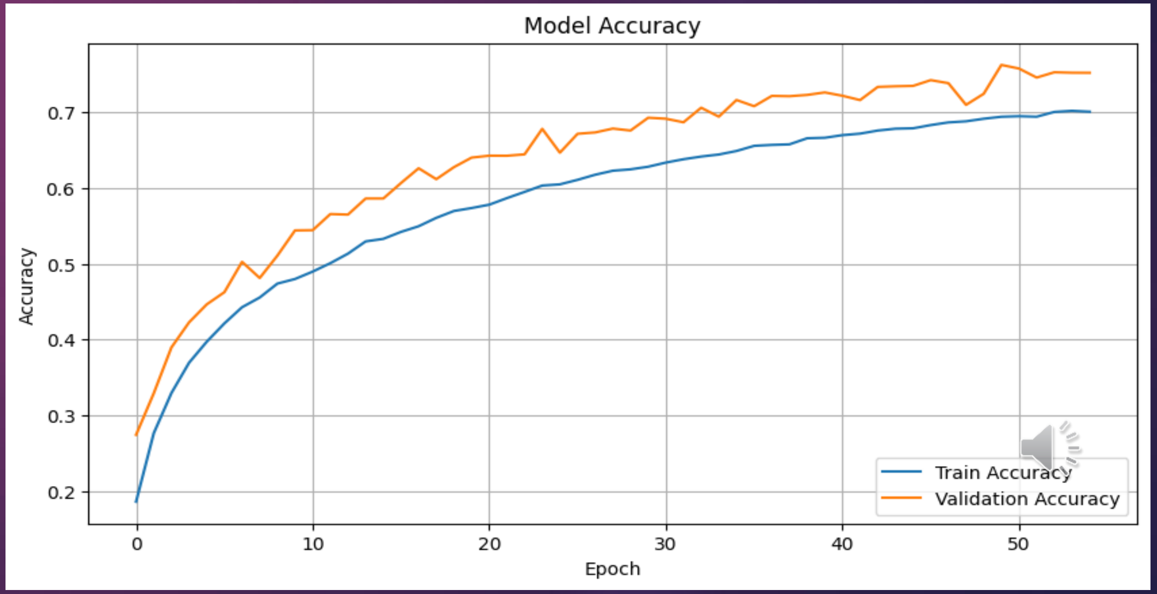
Outcome: Careful parameter tuning ensured stability, efficiency, and effective learning.

(Granziol, Zohren, and Roberts, 2022; Ji, Li, and Telgarsky, 2021; Smith, 2017)

Validation Strategy

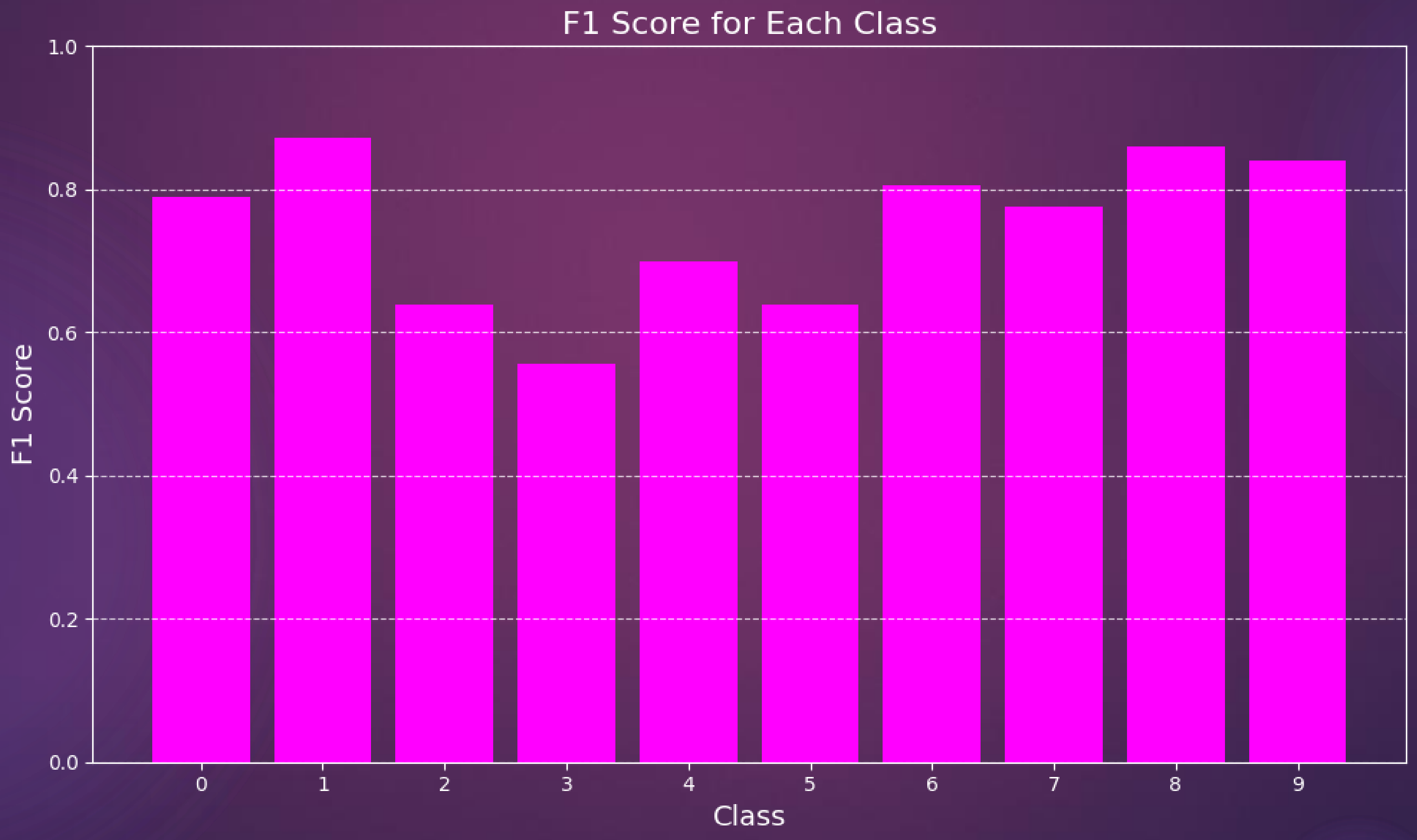
- Tracked validation accuracy and loss metrics following each epoch.
- Implemented early stopping to avoid overfitting and conserve resources.
- Retrieved the best weights to ensure optimal performance and strong generalisation.





(Mahsereci et al., 2017; Vilares Ferro et al., 2024)

Testing and Results



(Ahmadian et al., 2024; Bénédict et al., 2021; Tsoi et al., 2020)

Critical Analysis

Class Imbalances:

Precision and recall varied across classes, highlighting a need for targeted data collection or augmentation.

Overfitting:

Validation loss fluctuations suggest room for stronger regularisation.

Simple Architecture:

Deeper networks or pre-trained models could improve complex pattern recognition.

(Buda et al., 2018; Srivastava et al., 2014; He et al., 2016)

Reflections: Key Lessons and Growth

- Straightforward methods outperformed complex ones.
- Debugging required attention to detail.
- Early stopping and data augmentation improved optimization.
- Theory combined with practice enhanced learning.
- Resilience and adaptability drove progress

(Bergman et al., 2024; Schneider et al., 2021)

Bridging Theory and Real-World Impact

- Precision and recall inform critical systems.
- Data augmentation improves model performance in diverse conditions.
- Builds a foundation for scalable, adaptive AI solutions to tackle real-world challenges

(Chen et al., 2024; Tong et al., 2023)

Conclusion

This project was not just about achieving 75.62% accuracy—it was a journey of creativity, growth, and problem-solving. Challenges became lessons, paving the way for exploring even greater possibilities in deep learning.

This is just the beginning of the adventure! 🚀

References

- Ahmadian, R., Ghatee, M., and Wahlström, J., (2024). *Superior Scoring Rules for Probabilistic Evaluation of Single-Label Multi-Class Classification Tasks.* arXiv preprint arXiv:2407.17697. Available at: https://arxiv.org/abs/2407.17697 (Accessed: 20 January 2025)

- Asadi, B. and Jiang, H., (2020). *On Approximation Capabilities of ReLU Activation and Softmax Output Layer in Neural Networks*. arXiv preprint arXiv:2002.04060. Available at: https://arxiv.org/abs/2002.04060 (Accessed: 20 January 2025).

- Bénédict, G., Koops, V., Odijk, D., and de Rijke, M., (2021). *sigmoidF1: A Smooth F1 Score Surrogate Loss for Multilabel Classification*. arXiv preprint arXiv:2108.10566. Available at: https://arxiv.org/abs/2108.10566 (Accessed: 20 January 2025)

- Bergman, E., Purucker, L., and Hutter, F., (2024). *Don't Waste Your Time: Early Stopping Cross-Validation*. arXiv preprint arXiv:2405.03389. Available at: https://arxiv.org/abs/2405.03389 (Accessed: 20 January 2025).

- Brownlee, J., (2019). *A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. Machine Learning Mastery*. Available at: https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/ (Accessed: 20 January 2025).

- Buda, M., Maki, A., and Mazurowski, M.A., (2018). *A systematic study of the class imbalance problem in convolutional neural networks.* Neural Networks, 106, pp.249–259. [online] Available at: https://arxiv.org/abs/1710.05381 (Accessed: 20 January 2025).

- Chen, H., Dan, L., Lu, Y., Chen, M., and Zhang, J., (2024). *An improved data augmentation approach and its application in medical named entity recognition*. BMC Medical Informatics and Decision Making, 24, Article number: 221. Available at: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-024-02624-x (Accessed: 20 January 2025).

- Dai, H., Peng, X., Shi, X., He, L., Xiong, Q. and Jin, H., (2022). *Reveal training performance mystery between TensorFlow and PyTorch in the single GPU environment*. Science China Information Sciences, 65(11), Article 112103. Available at: https://link.springer.com/article/10.1007/s11432-020-3182-1 (Accessed: 20 January 2025).

- Franke, M. and Degen, J., (2023). *The Softmax Function: Properties, Motivation, and Interpretation*. Available at: https://alpslab.stanford.edu/papers/FrankeDegen_submitted.pdf (Accessed: 20 January 2025).

- Granziol, D., Zohren, S., & Roberts, S., (2022). *Learning rates as a function of batch size: A random matrix theory approach to neural network training*. Journal of Machine Learning Research, 23(173), pp.1–65. Available at: https://jmlr.org/papers/v23/20-1258.html (Accessed: 20 January 2025).

- He, K., Zhang, X., Ren, S., and Sun, J., (2016). *Deep residual learning for image recognition.* In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–778. Available at: https://arxiv.org/abs/1512.03385 (Accessed: 20 January 2025).

- Hebart, M.N. and Vicente, R., (2016). *An Efficient Data Partitioning to Improve Classification Performance While Keeping Parameters Interpretable*. PLOS ONE, 11(8), p.e0161788. Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0161788 (Accessed: 20 January 2025)

- Huang, L., Yang, D., Luo, Z., Xu, D. and Lu, H., (2020). *Normalization Techniques in Training Deep Neural Networks: A Survey*. arXiv preprint arXiv:2009.12836. Available at: https://arxiv.org/pdf/2009.12836 (Accessed: 20 January 2025).

- Ji, Z., Li, J.D., and Telgarsky, M., (2021). Early-stopped neural networks are consistent. *Advances in Neural Information Processing Systems*, 34, pp. 1–13. Available at: https://proceedings.neurips.cc/paper/2021/file/0e1ebad68af7f0ae4830b7ac92bc3c6f-Paper.pdf (Accessed: 20 January 2025).

- Krizhevsky, A. & Hinton, G., (2009). *Learning multiple layers of features from tiny images*. Available at: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (Accessed 20 January 2025)

- Krizhevsky, A., Sutskever, I., & Hinton, G.E., (2012). *ImageNet classification with deep convolutional neural networks*. Advances in Neural Information Processing Systems, 25, pp.1097–1105. Available at: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (Accessed: 20 January 2025).

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. Nature, 521(7553), pp. 436–444. Available at: https://doi.org/10.1038/nature14539 (Accessed: 20 January 2025)

- Mahsereci, M., Balles, L., Lassner, C., and Hennig, P., (2017). *Early Stopping Without a Validation Set. arXiv preprint arXiv:1703.09580.* Available at: https://arxiv.org/abs/1703.09580 (Accessed: 20 January 2025).
- Mao, A., Mohri, M. and Zhong, Y., (2023). *Cross-Entropy Loss Functions: Theoretical Analysis and Applications*. arXiv preprint arXiv:2304.07288. Available at: https://arxiv.org/abs/2304.07288 (Accessed: 20 January 2025).
- Medavarapu, S.S., (2024). *Advancements in Deep Learning: A Review of Keras and TensorFlow Frameworks*. Journal of Scientific and Engineering Research, 11(5), pp. 282–286. Available at: https://jsaer.com/download/vol-11-iss-5-2024/JSAER2024-11-5-282-286.pdf (Accessed: 20 January 2025).
- Murugan, P., (2018). *Implementation of Deep Convolutional Neural Network in Multi-class Categorical Image Classification*. arXiv preprint arXiv:1801.01397. Available at: https://arxiv.org/abs/1801.01397 (Accessed: 20 January 2025)
- Schneider, F., Dangel, F., and Hennig, P., (2021). *Cockpit: A Practical Debugging Tool for the Training of Deep Neural Networks. In: Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021).* Available at: https://proceedings.neurips.cc/paper/2021/hash/ae3539867aaeec609a4260c6feb725f4-Abstract.html (Accessed: 20 January 2025).
- Shorten, C. & Khoshgoftaar, T.M., (2019). *A survey on Image Data Augmentation for Deep Learning.* Journal of Big Data, 6(1), p.60. Available at: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0 (Accessed: 20 January 2025).
- Smith, L.N., (2017). *Cyclical learning rates for training neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 4641–4650. [online] Available at: https://arxiv.org/abs/1506.01186 (Accessed: 20 January 2025)
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R., (2014). *Dropout: A simple way to prevent neural networks from overfitting*. Journal of Machine Learning Research, 15(56), pp.1929–1958. Available at: http://jmlr.org/papers/v15/srivastava14a.html (Accessed: 20 January 2025).
- Tong, W., Xie, J., Li, T., Deng, H., Geng, X., Zhou, R., Yang, D., Dai, B., Lu, L., Li, H., (2023). *3D Data Augmentation for Driving Scenes on Camera*. In: Lecture Notes in Computer Science, 13807, pp. 45–56. Available at: https://link.springer.com/chapter/10.1007/978-981-97-8508-7_4 (Accessed: 20 January 2025).
- Tsoi, N., Candon, K., Li, D., Milkessa, Y., and Vázquez, M., (2020). Bridging the Gap: Unifying the Training and Evaluation of Neural Network Binary Classifiers. *arXiv preprint arXiv:2009.01367*. Available at: https://arxiv.org/abs/2009.01367 (Accessed: 20 January 2025).

- Vilares Ferro, M., Doval Mosquera, Y., Ribadas Pena, F.J., and Darriba Bilbao, V.M., (2024). Early Stopping by Correlating Online Indicators in Neural Networks. *arXiv preprint arXiv:2402.02513*. Available at: https://arxiv.org/abs/2402.02513 (Accessed: 20 January 2025)

- Wang, Z., Liu, K., Li, J., Zhu, Y. and Zhang, Y., (2024). *Various Frameworks and Libraries of Machine Learning and Deep Learning: A Survey.* Archives of Computational Methods in Engineering, 31, pp. 1–24. [online] Available at: https://link.springer.com/article/10.1007/s11831-018-09312-w (Accessed: 20 January 2025).

- Xu, Y., Goodacre, R., (2018(. *On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning*. Journal of Analysis and Testing, 2(3), pp.249–262. Available at: https://link.springer.com/article/10.1007/s41664-018-0068-2 (Accessed: 20 January 2025).