COMP309 Final Project

By Viyanka Moodley (300565283)

30/10/2024

Introduction

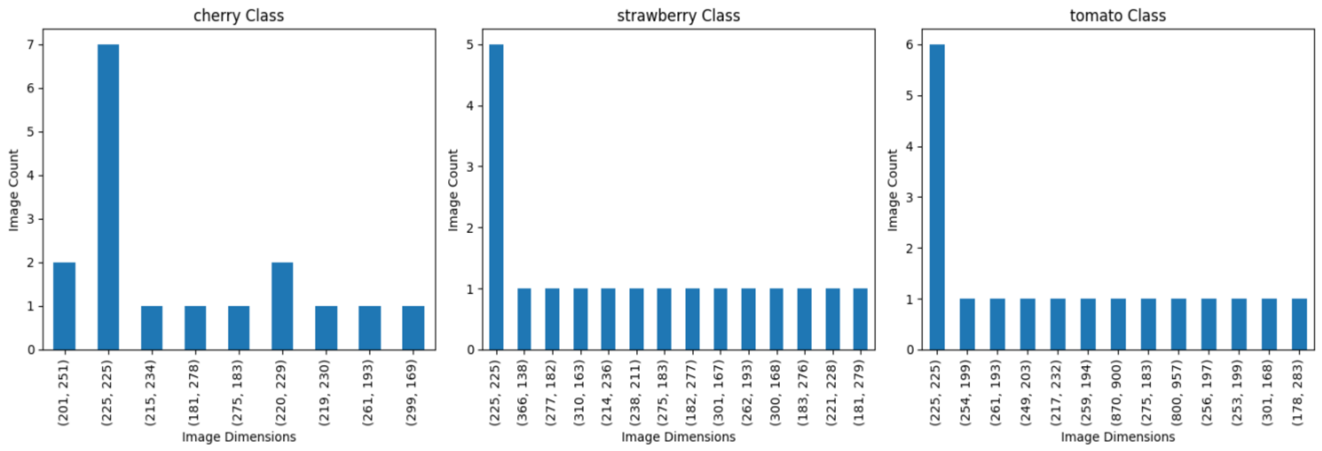
This study focuses on classifying images of three visually similar types of fruit - *cherries*, *strawberries*, and *tomatoes* using machine learning, specifically neural networks. The project aims to develop a model that can accurately distinguish between these categories, despite subtle visual differences, by evaluating two models: a baseline Multilayer Perceptron (MLP) and a Convolutional Neural Network (CNN). Accurate classification of visually similar objects has practical applications across industries such as agriculture, where automated sorting systems rely on precise image recognition. Given the visual similarities between these fruits in colour and shape, traditional image processing methods are less effective, and this study leverages the advanced feature extraction capabilities of deep learning to improve accuracy. The primary goal is to assess the effectiveness of MLP versus CNN for this task through several objectives: conducting exploratory data analysis (EDA) and applying preprocessing techniques like resizing, normalisation, and augmentation; developing and optimising both models; evaluating performance on metrics such as accuracy, loss, and training time; and performing a comparative analysis to understand each model's strengths and limitations. The findings are expected to highlight CNN's superior performance for this classification task, providing insights valuable to applications in automated sorting and quality control.

Problem Investigation

Exploratory Data Analysis (EDA)

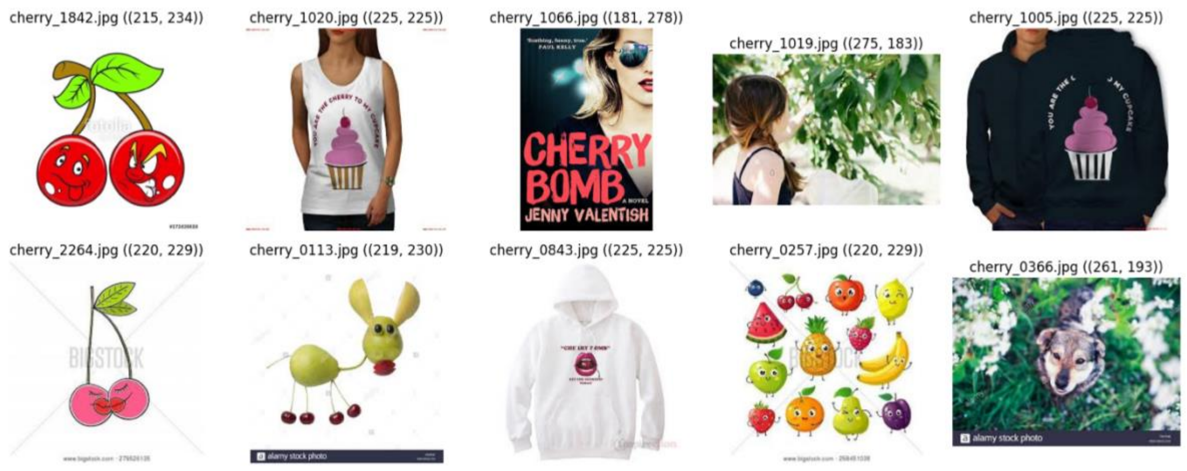
The initial dataset consisted of 4,485 images, with 1,495 images per class: *cherry*, *strawberry*, and *tomato*. A thorough exploratory data analysis (EDA) was conducted to identify potential inconsistencies and challenges within the data, focusing on colour channels, image dimensions, and file types.

Figure 1: Incorrect Image Dimensions in Each Class



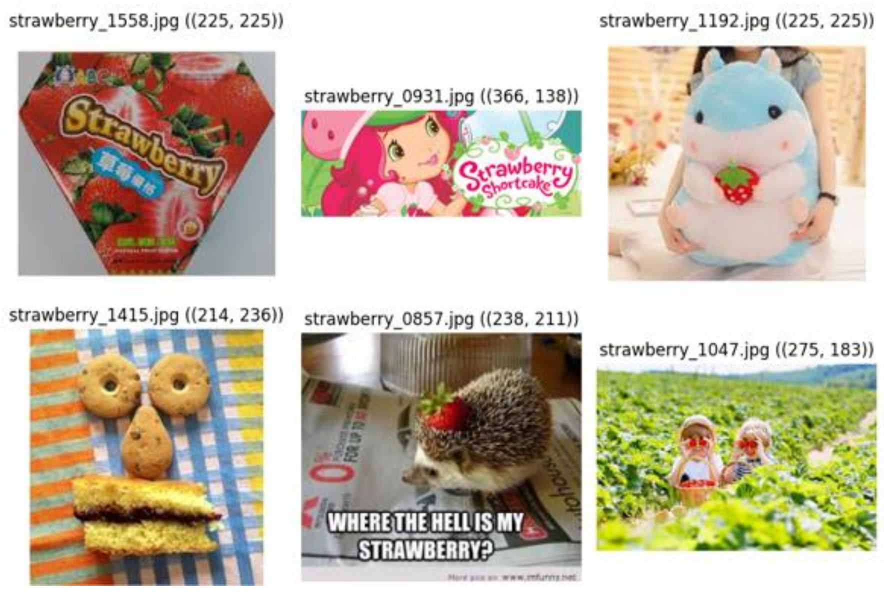
It was observed that approximately 18 images in each class deviated from the standard dimension of 300 by 300 pixels. Additionally, a subset of images used non-standard colour channels, specifically "RGBA" (RGB with an alpha transparency channel) and "L" (grayscale), rather than the expected "RGB" format.

Figure 2: A subset of images from the "cherry" class


cherry_1842.jpg ((215, 234))
cherry_1020.jpg ((225, 225))
cherry_1066.jpg ((181, 278))
cherry_1019.jpg ((275, 183))
cherry_1005.jpg ((225, 225))
cherry_2264.jpg ((220, 229))
cherry_0113.jpg ((219, 230))
cherry_0843.jpg ((225, 225))
cherry_0257.jpg ((220, 229))
cherry_0366.jpg ((261, 193))
CHERRY BOMB
JENNY VALENTISH

Visual inspection of images that did not conform to the standard dimensions revealed several issues; many of these images lacked the intended fruit, depicted multiple fruit types, or were incorrectly labelled. For instance, certain images contained textual representations of the fruit's name, such as the animated character "Strawberry Shortcake" or cherry blossom trees labelled as "cherry." Given their irrelevance and potential to introduce noise into the model, these non-standard images were excluded from the dataset to maintain data integrity and ensure that the dataset remained focused on the classification task at hand.

Figure 3: A subset of images from the "strawberry" class


strawberry_1558.jpg ((225, 225))
strawberry_0931.jpg ((366, 138))
strawberry_1192.jpg ((225, 225))
strawberry_1415.jpg ((214, 236))
strawberry_0857.jpg ((238, 211))
strawberry_1047.jpg ((275, 183))
Strawberry Shortcake
WHERE THE HELL IS MY STRAWBERRY?

Data Preprocessing and Feature Design

1. **Image Resizing and Filtering:** To ensure consistency across all samples, all images were resized to a standardised dimension of 75 by 75 pixels. This resizing step not only facilitated uniformity but also reduced computational demands during model training, aligning with established practices in computer vision that recommend downscaling images to improve processing efficiency without compromising classification performance (Russakovsky et al., 2015). Additionally, images with dimensions other than 300 by 300 pixels were filtered out based on the findings from EDA. This filtering was intended to enhance the model's capacity to learn relevant spatial patterns, unencumbered by irrelevant or mislabelled data that could degrade model accuracy.
2. **Colour Channel Conversion:** Given the presence of images in non-RGB colour channels, a custom preprocessing function was developed to convert all images to the RGB colour space. This step ensured that each image contained consistent colour data, which is essential for models relying on colour information for classification accuracy. Standardising all images to RGB format aligns with best practices in image preprocessing, as it mitigates discrepancies in colour representation and enhances data uniformity, a critical factor for neural networks sensitive to colour information (O'Shea & Nash, 2015).
3. **Data Augmentation:** To enhance the robustness and generalisability of the model, data augmentation techniques were applied to the training dataset. The augmentations included random horizontal flips, rotations, colour jitter, and random cropping. Data augmentation is a well-established method for artificially expanding a dataset, introducing variations that simulate different environmental conditions and orientations (Shorten & Khoshgoftaar, 2019). By applying these augmentations, the dataset was enriched, making the model more resilient to variations in colour and positioning, thus reducing overfitting and improving performance on unseen test data. This approach leverages the inherent robustness of convolutional neural networks (CNNs) by training them on diverse representations of each class, which can improve accuracy and reduce model bias.
4. **Feature Selection:** Explicit feature engineering was not performed in this study, as CNNs are capable of automatically learning hierarchical features from raw pixel data (LeCun, Bengio, & Hinton, 2015). Beyond resizing and normalisation, no additional feature extraction was conducted, allowing the model to autonomously identify and learn relevant features through its convolutional layers.

The preprocessing steps implemented based on EDA findings were instrumental in enhancing model accuracy, stability, and efficiency. First, converting images to RGB colour space eliminated inconsistencies in colour channels, ensuring that all images provided uniform colour information to the model. Second, resizing and filtering non-standard images contributed to a standardised dataset, facilitating the CNN's ability to learn consistent spatial patterns. Finally, data augmentation introduced controlled variations, which have been shown to improve generalisation by reducing overfitting in deep learning models (Shorten & Khoshgoftaar, 2019). Overall, these preprocessing steps strengthened the quality of the dataset, directly contributing to the improved performance of the classification model.

Methodology

This section details the steps undertaken to train and optimise the Convolutional Neural Network (CNN) model for the classification task, covering data handling, model architecture, training strategies, and optimisation methods.

Data Splitting and Cross-Validation

To develop a robust model, the dataset was split into training and validation subsets with an 80:20 ratio, a standard practice to prevent overfitting and assess model generalisation capabilities (Goodfellow, Bengio, & Courville, 2016). In addition to this split, 5-fold cross-validation was employed, where the data is partitioned into five subsets, with each subset used as the validation set in turn. Cross-validation provided a reliable performance estimate by exposing the model to various subsets, thus reducing the likelihood of overfitting to any single dataset split. This method ensured a more accurate assessment of model performance across different data distributions.

Model Architecture and Configuration

The CNN model was chosen for its ability to capture spatial hierarchies in images, making it highly suitable for tasks involving subtle visual distinctions between classes (O'Shea & Nash, 2015). The model architecture consisted of three convolutional layers, each followed by batch normalisation, ReLU activation, and max-pooling. The use of three progressively deeper convolutional layers enabled the model to extract increasingly complex features, from edges and textures in the initial layers to high-level class-specific features in the deeper layers. The final layers consisted of fully connected layers with dropout regularisation to mitigate overfitting and enhance generalisation. This architecture balances complexity and computational efficiency, ensuring the model has sufficient capacity to learn without excessive overfitting.

Loss Function

Cross-entropy loss was employed as the primary loss function, as it is widely used for multi-class classification tasks and measures the difference between predicted and actual labels effectively (Murphy, 2012). Cross-entropy loss is particularly suitable for categorical data, penalising the model for incorrect predictions while rewarding accurate classification. This function provided a reliable metric for model optimisation, allowing the model to focus on reducing prediction error.

Optimisation Method

Three optimisation methods were trialled during model training: **Stochastic Gradient Descent (SGD) with Momentum**, **Adam**, and **AdamW**. Initially, SGD with momentum was applied to accelerate convergence and avoid oscillations. However, while effective in initial testing, it did not yield optimal results. Subsequently, **Adam** (Adaptive Moment Estimation), an adaptive learning rate method, was tested for its capability to adjust learning rates dynamically for each parameter, improving stability and convergence (Kingma & Ba, 2014). The final optimisation method, **AdamW**, which combines Adam with weight decay regularisation, demonstrated superior performance, achieving a balance between convergence speed and overfitting mitigation. AdamW outperformed other optimisers in terms of validation accuracy and stability, making it the preferred choice for this classification task.

Regularisation Strategies

To prevent overfitting and improve model generalisation, two regularisation techniques were implemented:

1. **Dropout Regularisation**: A dropout rate of 0.5 was applied in the fully connected layers, randomly deactivating neurons during training to reduce co-adaptation of neurons and force the network to learn robust features (Srivastava et al., 2014). Dropout is a proven technique in neural networks for reducing overfitting, particularly in models with fully connected layers where overfitting is more common.
2. **Label Smoothing**: Label smoothing with a factor of 0.1 was used to prevent the model from becoming overconfident in its predictions by distributing a small portion of probability across all classes rather than assigning a probability of 1 to the true class. This technique has been shown to improve model calibration and reduce overfitting, leading to better generalisation (Szegedy et al., 2016).

Activation Function

ReLU (Rectified Linear Unit) was used as the activation function for all hidden layers due to its computational efficiency and ability to introduce non-linearity, allowing the model to learn complex patterns in the data. ReLU is widely preferred in CNNs as it mitigates the vanishing gradient problem, ensuring more effective gradient propagation during backpropagation (Nair & Hinton, 2010).

Hyperparameter Settings

Through experimentation, the following hyperparameters were determined to yield optimal results:

- **Learning Rate**: A learning rate of 0.001 was chosen for AdamW, balancing convergence speed with stability. This learning rate allowed for steady improvement without overshooting during gradient descent.
- **Batch Size**: 64 images per batch were used, which provided a balance between computational efficiency and effective gradient estimates.
- **Number of Epochs**: The model was trained for 10 epochs, sufficient for convergence without excessive training that could lead to overfitting.
- **Dropout Rate**: Set to 0.5 in the fully connected layers to reduce overfitting.

Data Augmentation

To further enrich the training dataset and enhance model robustness, various data augmentation techniques were applied to the images. These included random horizontal flipping, rotation, colour jitter, and random cropping. Data augmentation is a standard method to increase the effective size of the training dataset and simulate variations that the model may encounter during deployment (Shorten & Khoshgoftaar, 2019). By augmenting the data, the model became more resilient to positional and colour based changes, thus reducing overfitting and improving generalisation on unseen data.

Summary and Discussion

This study involved training and evaluating two neural network architectures, a baseline Multilayer Perceptron (MLP) and an optimised Convolutional Neural Network (CNN), to classify images of cherries, strawberries, and tomatoes. The MLP model consisted of a simple structure with one hidden layer of 512 neurons using ReLU activation, optimised with cross-entropy loss and the Adam optimiser. Conversely, the CNN model was designed with three convolutional layers, each followed by batch normalisation, ReLU activation, and max-pooling. Additional techniques such as dropout, label smoothing, and data augmentation (random flips, rotations, and colour jitter) were applied to improve the CNN's performance and generalisation.

Comparative Analysis of Results

Training Time and Convergence

The MLP model, with its simpler structure, had shorter training times per epoch but showed limited learning capacity, evident in its relatively high and slowly decreasing training and validation losses. The CNN, although more computationally intensive due to its complex architecture, demonstrated faster convergence and consistent accuracy improvements across epochs. This contrast in convergence rates reflects the CNN's capacity to learn intricate spatial patterns within image data, which the MLP, with its fully connected architecture, could not effectively capture.

Classification Performance

The final CNN substantially outperformed the MLP in terms of validation accuracy, achieving over 81% accuracy by the final epoch compared to the MLP's peak of around 52%. The accuracy comparison plot (Validation Accuracy Comparison) illustrates the CNN’s steady upward trend in accuracy, whereas the MLP plateaued early, suggesting it was unable to learn features critical for distinguishing between visually similar fruit classes.

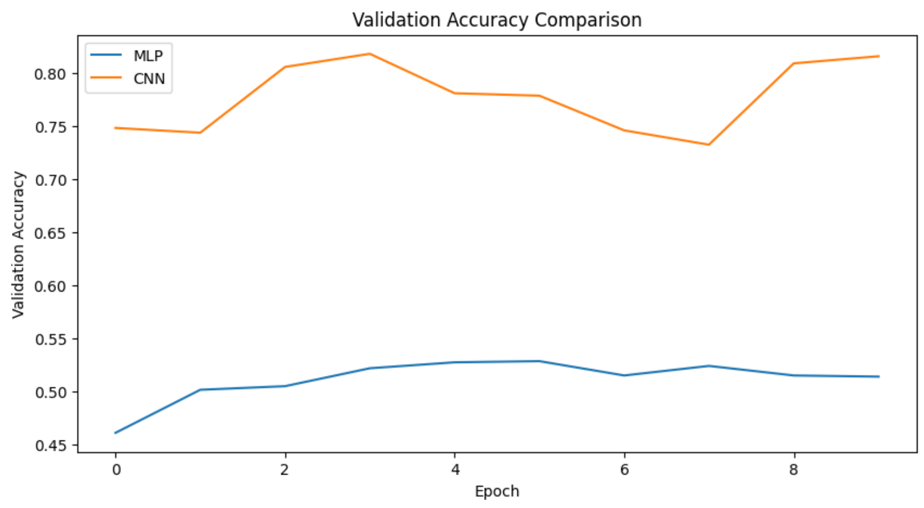


Figure 4: Validation Accuracy Comparison

Likewise, the validation loss plot (Validation Loss Comparison) shows a pronounced difference. While the CNN’s validation loss decreased consistently, stabilising at a low value, the MLP’s validation loss plateaued at a much higher level, indicating limited capacity to generalise. The CNN’s lower validation loss underscores its ability to model complex visual structures in the images, a crucial factor for achieving high classification accuracy.

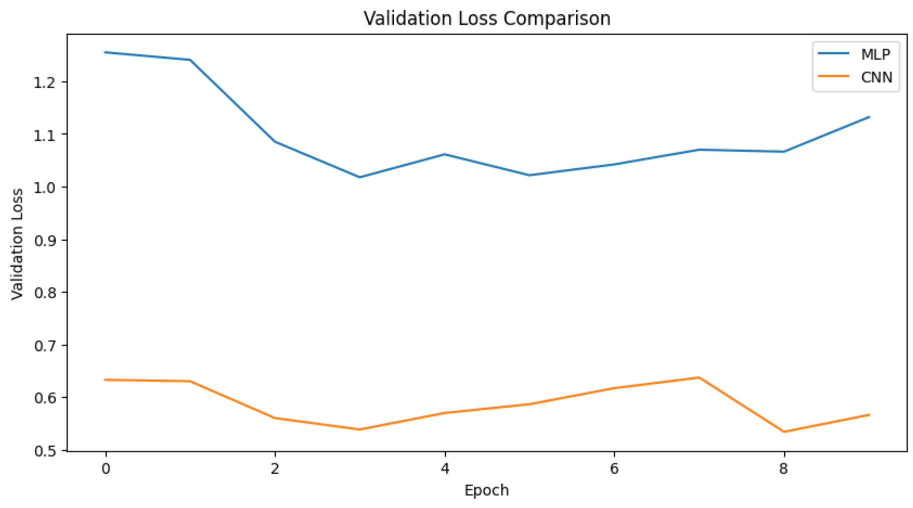


Figure 5: Validation Loss Comparison

Key Differences and Their Causes

One striking difference between the two models was the CNN’s significantly higher validation accuracy and lower validation loss compared to the MLP. This disparity is primarily attributed to the CNN's use of convolutional layers, which are specifically designed to capture spatial hierarchies and local features within images (LeCun et al., 2015). Convolutional layers act as filters that identify patterns, such as edges and textures, which are fundamental for distinguishing objects in image data. By stacking multiple convolutional layers, the CNN can progressively learn more complex features, allowing it to differentiate between cherries, strawberries, and tomatoes even when they share similar colour and shape.

In contrast, the MLP's fully connected architecture lacks this spatial awareness, as it treats each

Figure 6: Training Loss Comparison

pixel independently, losing the positional and relational information necessary for robust image classification. Consequently, the MLP was unable to effectively capture the spatial patterns that distinguish each fruit class, leading to its lower accuracy and higher loss.

Another notable difference was the impact of data augmentation and regularisation on the CNN. Techniques like random flipping, rotation, colour jitter, and dropout allowed the CNN to generalise better by exposing it to variations in the training data and simulating real-world conditions. These methods reduced overfitting, a common issue in deep learning, allowing the CNN to perform well on validation data. The MLP, however, did not benefit as much from these techniques, as its architecture did not have the capacity to leverage augmented data in the same way the CNN did. Additionally, label smoothing helped the CNN avoid overconfident predictions, which improved its stability and accuracy on unseen data, whereas the MLP showed little benefit from these techniques.

In summary, the CNN's superior architecture and regularisation methods enabled it to capture the nuanced differences among fruit classes, resulting in significantly better performance compared to the MLP. These findings underscore the critical role of model architecture in tasks requiring spatial feature extraction. The CNN's ability to process and learn from spatial relationships within the images allowed it to classify with much higher accuracy, making it a more suitable choice for image-based classification tasks.

Conclusions and Future Work

This study demonstrated the effectiveness of Convolutional Neural Networks (CNNs) over Multilayer Perceptrons (MLPs) for image classification tasks, specifically in distinguishing visually similar fruit classes: cherry, strawberry, and tomato. The final CNN model significantly outperformed the baseline MLP, achieving higher validation accuracy and lower validation loss. The CNN's architecture, with its convolutional layers capable of capturing spatial hierarchies and feature dependencies within images, proved to be a decisive factor in its success. Data augmentation and regularisation techniques, including dropout, label smoothing, and transformations such as random flips and rotations, further enhanced the CNN's robustness and ability to generalise to unseen data. In contrast, the MLP's lack of spatial feature extraction capability limited its performance, making it less suited for tasks involving nuanced visual differences.

The advantages of the CNN approach include its ability to automatically learn hierarchical features from raw pixel data, enabling it to classify complex image patterns with minimal manual feature engineering. The primary drawback of the CNN, however, is its higher computational cost and training time due to the complexity of its architecture. This may limit its scalability for larger datasets or applications with resource constraints. The MLP, while computationally efficient, is limited by its fully connected structure, making it unsuitable for tasks requiring spatial awareness.

Future Work

There are several promising directions to build on the current model's performance and usability. One approach is to apply transfer learning by fine-tuning pre-trained models like VGG16, ResNet, or Inception. These models, trained on large datasets such as ImageNet, bring powerful feature extraction capabilities that could significantly boost accuracy and reduce training time for our specific task. Another strategy is ensemble learning, where multiple CNN architectures are combined to improve classification performance. By blending the strengths of different models, ensemble techniques like averaging or stacking could make the classifier more resilient and accurate.

Adding attention mechanisms within the CNN architecture is another exciting possibility. This would allow the model to focus on the most relevant parts of each image, such as colour, texture, and shape, which is especially useful when distinguishing visually similar classes. For real-time applications, optimising the model for faster inference is essential; techniques like quantisation, pruning, or knowledge distillation could help reduce computational demands, making the model more practical for use on edge devices in settings like automated sorting systems. Finally, expanding the dataset to include images with diverse backgrounds, lighting conditions, and viewpoints would improve the model's robustness, enabling it to generalise better in real-world situations. Together, these enhancements would help refine the model’s accuracy, efficiency, and adaptability, moving it closer to real-world deployment in industrial applications.

References

1. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *In 2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248-255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848

2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
3. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. https://doi.org/10.48550/arXiv.1412.6980

4. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

5. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. *In Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (pp. 807-814). https://www.cs.toronto.edu/~hinton/absps/reluICML.pdf

6. O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*. https://doi.org/10.48550/arXiv.1511.08458

7. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211-252. https://doi.org/10.1007/s11263-015-0816-y

8. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 1-48. https://doi.org/10.1186/s40537-019-0197-0

9. 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*(1), 1929-1958. https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

10. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2818-2826). IEEE. https://doi.org/10.1109/CVPR.2016.308