Designing and Regularizing Deep CNN Architectures for Dog versus Cat Image Classification

Athanasios Kanavos[1], Gerasimos Vonitsanos[2], Manolis Maragoudakis[3] and Phivos Mylonas[4]

[1] Department of Information and Communication Systems Engineering, University of the Aegean, Samos, Greece
icsdd20017@icsd.aegean.gr
[2] Computer Engineering and Informatics Department University of Patras, Patras, Greece
mvonitsanos@ceid.upatras.gr
[3] Department of Informatics, Ionian University, Corfu, Greece
mmarag@ionio.gr
[4] Department of Informatics and Computer Engineering University of West Attica, Athens, Greece
mylonasf@uniwa.gr

Abstract. Convolutional Neural Networks (CNNs) have become fundamental to modern computer vision, delivering state-of-the-art results in a wide range of image classification tasks. This study investigates the binary classification of dog and cat images using custom-designed CNN architectures, trained entirely from scratch without leveraging pretrained weights. Three distinct models are developed, varying in depth, convolutional block structure, and regularization techniques, to systematically evaluate the impact of architectural design choices on classification performance. Preprocessing strategies, including data augmentation and pixel value normalization, are employed to enhance model robustness and prevent overfitting. Extensive experimental evaluation on the Dogs vs. Cats dataset demonstrates that deeper architectures, when combined with batch normalization and dropout, achieve superior generalization and convergence behavior. The best-performing model attains a validation accuracy of approximately 92%, confirming the effectiveness of the proposed approach. These findings underscore the importance of thoughtful CNN design and regularization, offering valuable insights for future applications in automated animal recognition and real-world visual classification systems.

Keywords: Convolutional Neural Networks · Deep Learning · CNN Architecture Design · Regularization Techniques · Image Classification · Data Augmentation · Computer Vision

1 Introduction

Image classification is a fundamental task in computer vision, underpinning a wide range of applications such as medical diagnosis, autonomous driving, security surveillance, and animal species identification [24,27]. Traditional approaches relied heavily on handcrafted features and classical machine learning algorithms, including Support Vector Machines (SVMs) and Random Forests, which demanded extensive feature engineering and domain-specific expertise [23]. The advent of deep learning, particularly Convolutional Neural Networks (CNNs) [13,21], has transformed this landscape by enabling automatic feature extraction and hierarchical representation learning directly from raw images [16]. CNNs have achieved remarkable success in large-scale challenges such as ImageNet [15] and have become the backbone of modern computer vision systems [12,20].

A widely studied benchmark in binary image classification is the differentiation between dogs and cats. Although the task appears straightforward, it poses significant challenges due to variability in poses, fur textures, lighting conditions, and occlusions [28]. The Dogs vs. Cats dataset, introduced by Kaggle, comprises 25,000 labeled images and serves as a standard testbed for evaluating CNN architectures [6]. Its balanced nature and visual complexity make it ideal for assessing feature extraction capabilities and generalization in binary classification tasks.

Over the years, CNN architectures have evolved significantly, leading to seminal models such as AlexNet [15], VGGNet [28], ResNet [6], and EfficientNet [30]. These architectures utilize convolutional layers to capture spatial hierarchies, pooling layers for dimensionality reduction, and fully connected layers for final classification [2]. In this study, we design and implement CNN architectures from scratch, deliberately avoiding reliance on pretrained models, to evaluate the intrinsic feature learning capabilities of CNNs. Various architectural and training hyperparameters, including the number of convolutional layers, filter sizes, activation functions, and optimization strategies, are systematically explored to optimize model performance.

Data preprocessing and augmentation are critical for enhancing generalization and mitigating overfitting in deep learning models [3]. In this work, we apply augmentation techniques such as random rotations, flipping, and contrast adjustments [27], along with pixel value normalization [9], to stabilize training dynamics and improve model robustness. Model performance is rigorously evaluated using key metrics such as accuracy, precision, recall, and F1-score [23], providing a comprehensive assessment of model effectiveness.

The main contributions of this work are as follows. First, we design and implement three custom CNN architectures specifically tailored for binary image classification without relying on pretrained networks. Second, we systematically investigate the impact of architectural depth, batch normalization, dropout, and data augmentation on model generalization and convergence. Third, we conduct an extensive experimental evaluation using multiple performance metrics, providing insights into the optimization of CNN architectures for small- to medium-scale datasets.

The remainder of the paper is organized as follows. Section 2 reviews related work in CNN-based image classification and highlights recent advances. Section 3 presents the fundamental concepts underpinning our approach, including convolutional layers, pooling layers, batch normalization, and dropout. Section 4 describes the proposed CNN architectures, detailing their structural variations and regularization strategies. Section 5 reports the experimental setup, dataset characteristics, and comparative evaluation results. Finally, Section 6 concludes the paper and outlines future research directions.

2 Related Work

The field of image classification has witnessed significant advancements driven by the development of CNNs. CNNs have proven highly effective for automatic feature extraction and classification across a wide range of computer vision tasks, consistently outperforming traditional machine learning approaches based on handcrafted features [10,16]. A major breakthrough was achieved with AlexNet, which won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, substantially reducing classification error rates compared to conventional methods [15]. This success catalyzed the development of deeper architectures, such as VGGNet [28], ResNet [6], and EfficientNet [30], each introducing innovations aimed at improving accuracy and computational efficiency.

Within the domain of binary image classification, the Dogs vs. Cats problem has emerged as a widely studied benchmark task. Researchers have explored a variety of CNN-based models for this challenge, experimenting with different architectural depths, hyperparameter tuning, and regularization strategies to enhance classification performance [7,19,27]. Early approaches predominantly utilized shallow networks with limited layers, whereas more recent efforts have incorporated deeper CNNs, and transfer learning methodologies to boost feature extraction and decision-making capabilities [9]. In particular, transfer learning with pretrained models such as VGG16 and ResNet50 has been shown to deliver high accuracy with relatively low computational overhead [23].

A persistent challenge in deep learning for image classification is overfitting, where models tend to memorize training data rather than generalize to unseen samples [25]. To counter this, a range of regularization techniques has been proposed, including dropout [29], batch normalization [9], and extensive data augmentation [27]. Data augmentation methods, such as random rotations, horizontal flipping, and color perturbations, have been widely adopted to synthetically increase dataset variability, thereby improving model generalization. Furthermore, optimization algorithms like Adam [14] have been introduced to accelerate convergence and enhance training stability.

Recent research has also focused on developing lightweight CNN architectures capable of achieving high accuracy with reduced computational complexity. Models such as MobileNet [8] and ShuffleNet [31] employ depthwise separable convolutions and grouped convolutions, respectively, to significantly lower the number of parameters while preserving performance, facilitating deployment in

resource-constrained environments. These advancements underscore the ongoing evolution of CNN design, aimed at balancing accuracy, efficiency, and scalability.

Building upon this extensive body of work, the present study proposes the development of custom CNN architectures specifically designed for the Dogs vs. Cats classification task. Distinct from transfer learning approaches that utilize pretrained networks, our work focuses on training CNNs from scratch, systematically investigating the effects of architectural depth, regularization techniques, and data augmentation strategies on classification performance and convergence behavior.

3 Methodology Foundations

A clear understanding of the key components of Convolutional Neural Networks (CNNs) is essential for designing effective deep learning models for image classification tasks. This section briefly reviews the fundamental concepts and operations relevant to the proposed architectures, including convolutional layers, pooling layers, batch normalization, and dropout, which are integral to the CNN architectures proposed in this study.

3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are the foundation of modern computer vision, enabling automatic hierarchical feature extraction from raw image data. Early layers detect simple patterns such as edges and textures, while deeper layers capture more complex structures like shapes and object parts. CNNs are particularly suited for image classification tasks due to their ability to learn spatial hierarchies while maintaining computational efficiency [5]. Typical CNNs consist of convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for final classification.

3.2 Convolutional Layers

Convolutional layers apply learnable filters across input images to generate feature maps, extracting spatial features essential for classification. Each convolution operation preserves local spatial relationships while reducing the number of parameters compared to fully connected layers. Key parameters such as filter size, stride, and padding influence the output dimensions and the receptive field of extracted features.

3.3 Pooling Layers

Pooling layers, particularly max pooling, reduce the spatial dimensions of feature maps while retaining salient features. By selecting the maximum activation within pooling windows, max pooling improves model robustness to small input variations and reduces computational complexity, supporting efficient and generalizable feature extraction.

3.4 Batch Normalization

Batch Normalization (BN) stabilizes and accelerates training by normalizing layer inputs to have zero mean and unit variance within each mini-batch. This reduces internal covariate shift, enables higher learning rates, and improves generalization. BN has become a standard component in deep CNN architectures to enhance convergence and performance [4,9].

3.5 Dropout

Dropout is a regularization technique that randomly deactivates neurons during training, preventing overfitting by encouraging distributed feature learning. By training multiple implicit sub-networks, dropout improves model robustness and generalization to unseen data [29].

4 Proposed Architectures

To develop an effective and robust deep learning model for binary image classification, we propose three Convolutional Neural Network (CNN) architectures specifically designed for the Dogs vs. Cats dataset. CNNs are particularly well-suited for visual recognition tasks, as they automatically learn spatial hierarchies of features from raw image data, reducing the need for manual feature engineering. The proposed architectures differ in depth, the number of convolutional layers per block, and the application of regularization techniques such as batch normalization and dropout, enabling a comprehensive evaluation of how different design choices influence classification performance and generalization ability.

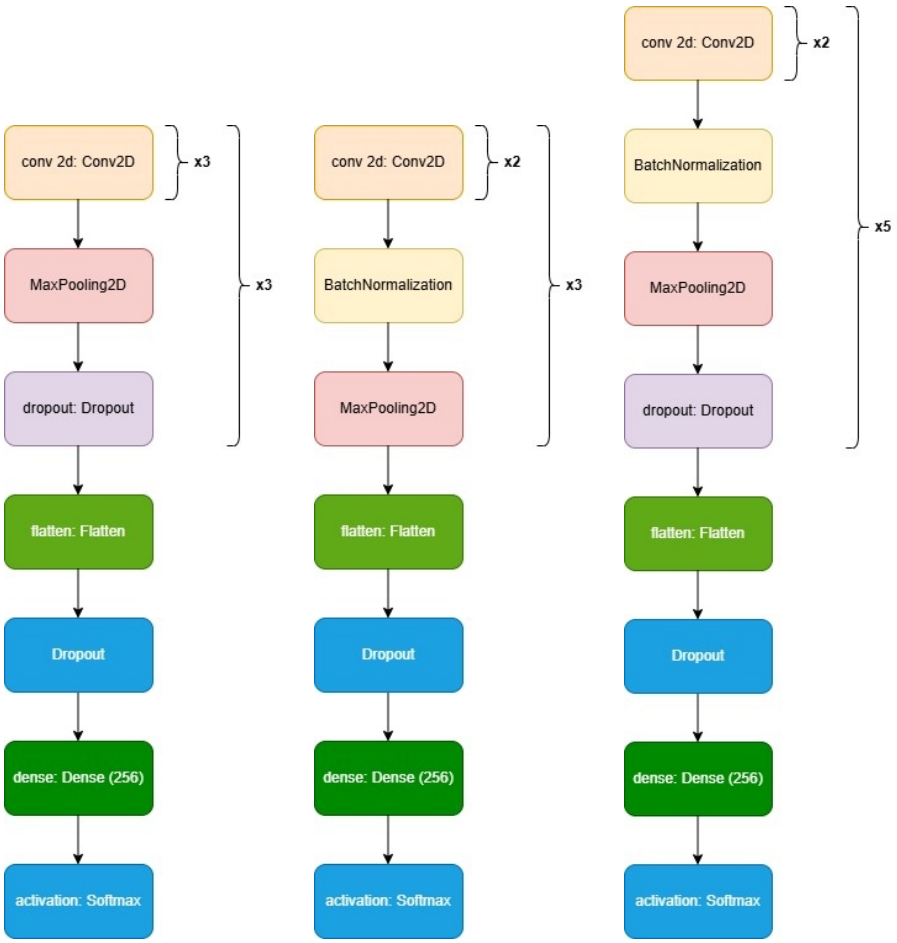
The first proposed architecture consists of three repeated convolutional blocks, each containing three consecutive convolutional layers followed by a max-pooling operation and a dropout layer. This deeper configuration aims to enable rich hierarchical feature extraction while mitigating overfitting. The second architecture introduces batch normalization after every pair of convolutional layers within each block and maintains a slightly shallower depth, comprising four convolutional blocks. The third architecture combines batch normalization and dropout within each convolutional block, extending the network depth to five blocks to enhance regularization and model robustness. The detailed layer configurations of the proposed CNN architectures are summarized in Table 1.

Figure 1 illustrates the structural differences among the proposed models, showing how convolutional, pooling, normalization, and dropout layers are organized to progressively extract and process features for binary classification.

Overall, the first and second architectures aim to capture intricate patterns and high-level abstractions from input images through deeper convolutional blocks, whereas the third architecture emphasizes balancing model complexity and robustness. By integrating extensive regularization mechanisms, the third model seeks to achieve high classification accuracy while mitigating the risks of overfitting. This diversity in architectural design enables a thorough analysis of how CNN depth and regularization strategies affect performance in binary image classification tasks.

Table 1. Layer Configurations of the Proposed CNN Architectures

Architecture	Layer Sequence and Operations
1st	(Conv2D × 3 → MaxPooling2D → Dropout) × 3 → Flatten → Dropout → Dense → Softmax
2nd	(Conv2D × 2 → BatchNormalization → MaxPooling2D) × 4 → Flatten → Dropout → Dense → Softmax
3rd	(Conv2D × 2 → BatchNormalization → MaxPooling2D → Dropout) × 5 → Flatten → Dropout → Dense → Softmax



(a) First Proposed CNN Architecture (b) Second Proposed CNN Architecture (c) Third Proposed CNN Architecture

Fig. 1. Layer configurations of the proposed CNN architectures

5 Evaluation

This section presents a comprehensive evaluation of the proposed CNN architectures using several performance metrics, including classification accuracy, loss convergence behavior, and computational efficiency. The evaluation was conducted on an independent test set to ensure a fair assessment of each model's generalization ability.

5.1 Dataset

The dataset used in this study is the Dogs vs. Cats dataset, sourced from Kaggle [1], containing a total of 10,000 labeled images evenly distributed between two classes: dogs and cats. Each image is labeled for binary classification. The dataset exhibits high variability in resolution, background complexity, and lighting conditions, enhancing its suitability for evaluating model robustness. All images were resized to 128×128 pixels, preserving aspect ratio when necessary. To augment data diversity and mitigate overfitting, random rotations, horizontal flips, and brightness adjustments were applied during training. The dataset was split into training (80%) and testing (20%) subsets to enable effective model evaluation. Table 2 summarizes the distribution across splits.

Table 2. Distribution of Data Instances Across Categories

Subset	**Dogs**	**Cats**	**Total Images**
Train	4000	4000	8000
Test	1000	1000	2000
Total	**5000**	**5000**	**10000**

5.2 Results

The validation accuracy of the three CNN architectures over 20 training epochs is illustrated in Figure 2. All models achieved steady improvements throughout training. The third architecture, which incorporated batch normalization and dropout, attained the highest validation accuracy of approximately 92%, followed by the second architecture at around 89% and the first architecture at approximately 85%. In addition to achieving the highest final accuracy, the third architecture also demonstrated a faster convergence rate, with accuracy increasing more sharply during the early epochs compared to the other models.

The validation loss curves for all models are presented in Figure 3. While all architectures successfully minimized loss during training, the deeper networks, particularly the third architecture, achieved smoother convergence and lower final loss values. Moreover, the third model exhibited a steadier decline in loss, with fewer fluctuations, indicating more consistent optimization across epochs.

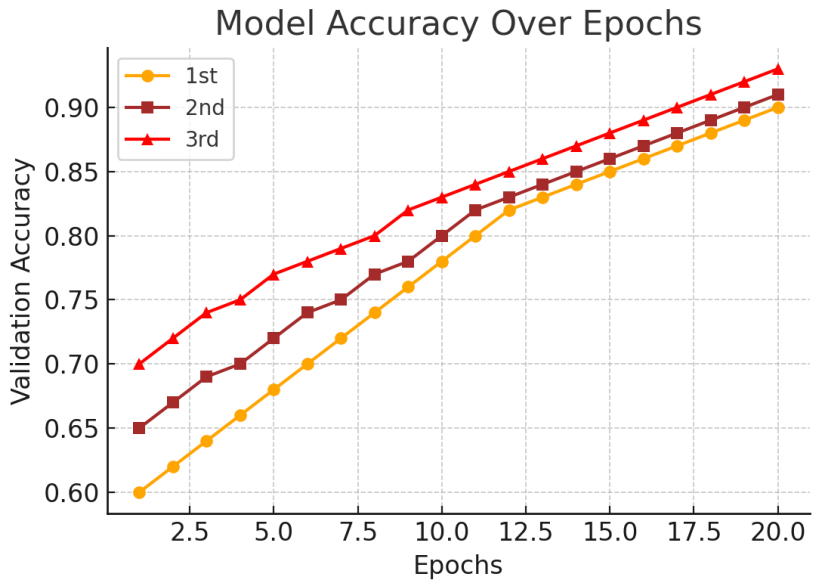


Fig. 2. Validation Accuracy over Epochs for the Three Proposed CNN Architectures

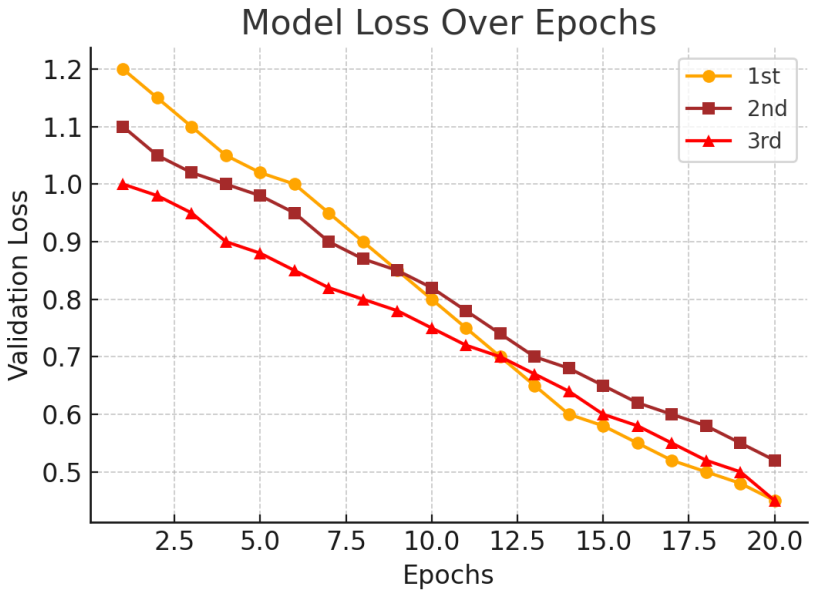


Fig. 3. Validation Loss over Epochs for the Three Proposed CNN Architectures

Training time was recorded for each architecture, as shown in Figure 4. The first model required the least training time (approximately 5 minutes), while

the third model required the most (about 12.5 minutes). This increase in computational cost correlates with model depth and complexity. Despite the higher training time, the third architecture achieved significantly better generalization, making the additional computational expense worthwhile in applications where classification accuracy is critical.

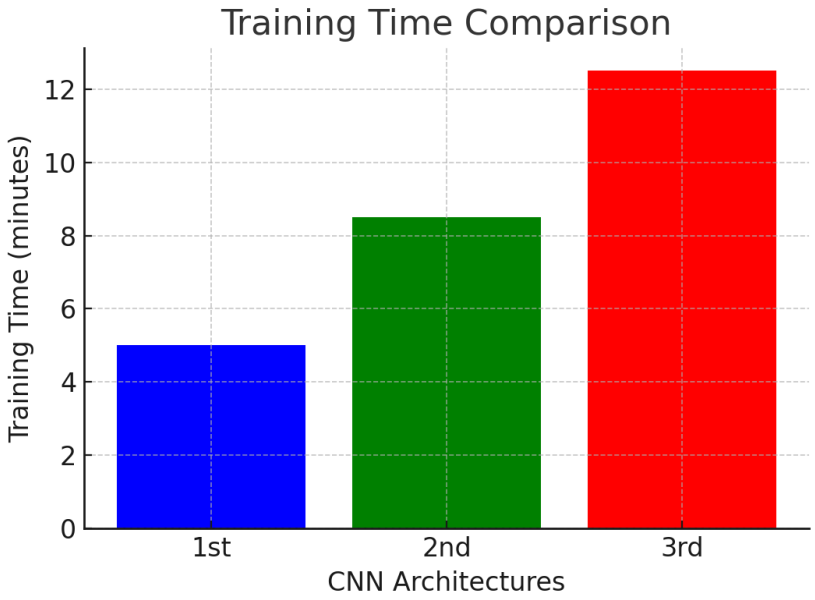


Fig. 4. Training time Comparison across the Three CNN Architectures

5.3 Discussion

The experimental results demonstrate that deeper CNN architectures, when enhanced with appropriate regularization techniques such as batch normalization and dropout, significantly improve binary image classification performance. The third architecture achieved the highest validation accuracy and the lowest validation loss, confirming that increasing network depth, when combined with effective regularization, enhances feature extraction and generalization ability.

Batch normalization contributed to stabilizing and accelerating the training process by normalizing layer activations, thus mitigating internal covariate shift and improving gradient flow. Dropout, by randomly deactivating neurons during training, effectively reduced overfitting, encouraging the model to learn more robust and distributed feature representations. Together, these regularization methods proved critical in enabling the deeper architecture to outperform its shallower counterparts.

Although deeper models incurred a computational cost in terms of increased training time, the relative improvement in classification performance justifies this overhead for applications where predictive accuracy is prioritized. Importantly, the gap in performance between the second and third architectures highlights that simply adding layers is not sufficient; careful architectural design and appropriate regularization are essential to fully leverage the benefits of deeper networks.

These findings underscore the importance of architectural depth and regularization not only for achieving higher accuracy but also for ensuring model stability and convergence. The third architecture provides a strong foundation for future extensions, including potential transfer learning applications, more advanced data augmentation strategies, or optimization techniques targeting further improvements in efficiency and scalability.

6 Conclusions and Future Work

This study presented the design, implementation, and evaluation of three Convolutional Neural Network (CNN) architectures for binary image classification on the Dogs vs. Cats dataset. By systematically varying network depth and integrating regularization techniques such as batch normalization and dropout, we explored the effects of architectural choices on model performance, convergence behavior, and computational efficiency. Data augmentation and normalization techniques were also employed to enhance model generalization and stability during training.

Experimental results demonstrated that deeper architectures, when properly regularized, significantly outperform shallower models, achieving higher validation accuracy and more stable convergence. The third proposed CNN, featuring both batch normalization and dropout across multiple convolutional blocks, attained a validation accuracy of approximately 92%, while maintaining smooth loss reduction and acceptable training times. These findings confirm the critical role of both architectural depth and regularization in developing robust CNN models for real-world image classification tasks.

Future work could focus on further refining the network architectures by exploring advanced techniques such as residual connections, dense connections, or attention mechanisms. Additionally, hyperparameter optimization strategies, including automated learning rate scheduling or adaptive optimizers, could be integrated to enhance convergence speed and final performance without substantially increasing computational cost [11,26].

Beyond technical improvements, future research could investigate the application of the proposed models to more complex or larger-scale datasets, such as ImageNet or domain-specific image collections [18,22]. Transfer learning approaches could also be explored, allowing the trained networks to adapt to different classification tasks with minimal retraining, thereby extending the practical applicability of the proposed designs to broader real-world scenarios [17].

In summary, this work highlights the importance of thoughtful architectural design and regularization in CNN-based image classification. The insights gained from this study provide a strong foundation for future enhancements and applications, paving the way for more efficient, accurate, and generalizable deep learning models in computer vision.

References

1. Dogs & cats images). `https://www.kaggle.com/datasets/chetankv/dogs-cats-images/data`, online; accessed on 28 April 2025
2. Aditi, Prasad, V.K., Gerogiannis, V.C., Kanavos, A., Dansana, D., Acharya, B.: Utilizing convolutional neural networks for resource allocation bottleneck analysis in cloud ecosystems. Cluster Computing **28**(1), 22 (2025)
3. Akuthota, S., Janapati, R.C., Kumar, K.R., Gerogiannis, V.C., Kanavos, A., Acharya, B., Grivokostopoulou, F., Desai, U.: Enhancing real-time cursor control with motor imagery and deep neural networks for brain–computer interfaces. Information **15**(11), 702 (2024)
4. Bjorck, J., Gomes, C.P., Selman, B., Weinberger, K.Q.: Understanding batch normalization. In: Annual Conference on Neural Information Processing Systems (NeurIPS). pp. 7705–7716 (2018)
5. Ghahnavieh, A.E., Luo, S., Chiong, R.: Deep learning to detect alzheimer's disease from neuroimaging: A systematic literature review. Computer Methods and Programs in Biomedicine **187**, 105242 (2020)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. IEEE Computer Society (2016)
7. Hindarto, D.: Use resnet50v2 deep learning model to classify five animal species. Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi) **7**(4), 758–768 (2023)
8. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017)
9. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: 32nd International Conference on Machine Learning (ICML). JMLR Workshop and Conference Proceedings, vol. 37, pp. 448–456. JMLR.org (2015)
10. Kanavos, A., Mylonas, P.: Deep learning analysis of histopathology images for breast cancer detection: A comparative study of resnet and VGG architectures. In: 18th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP). pp. 1–6. IEEE (2023)
11. Kanavos, A., Trigka, M., Dritsas, E., Vonitsanos, G., Mylonas, P.: A regularization-based big data framework for winter precipitation forecasting on streaming data. Electronics **10**(16), 1872 (2021)
12. Kanavos, A., Papadimitriou, O., Al-Hussaeni, K., Karamitsos, I., Maragoudakis, M.: Analyzing deep learning techniques in natural scene image classification. In: International Conference on Big Data (BigData). pp. 5682–5691. IEEE (2024)
13. Kanavos, A., Papadimitriou, O., Vonitsanos, G., Maragoudakis, M., Mylonas, P.: Advanced CNN architectures for improved garbage image classification: An in-depth evaluation. In: 9th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM). pp. 85–90. IEEE (2024)

14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations (ICLR) (2015)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: 26th Annual Conference on Neural Information Processing Systems. pp. 1106–1114 (2012)
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
17. Lyras, A., Vernikou, S., Kanavos, A., Sioutas, S., Mylonas, P.: Modeling credibility in social big data using LSTM neural networks. In: 17th International Conference on Web Information Systems and Technologies (WEBIST). pp. 599–606 (2021)
18. Mohanty, C., Mahapatra, S., Acharya, B., Kokkoras, F., Gerogiannis, V.C., Karamitsos, I., Kanavos, A.: Using deep learning architectures for detection and classification of diabetic retinopathy. Sensors **23**(12), 5726 (2023)
19. Moradi, R., Berangi, R., Minaei, B.: A survey of regularization strategies for deep models. Artificial Intelligence Review **53**(6), 3947–3986 (2020)
20. Papadimitriou, O., Kanavos, A., Maragoudakis, M., Gerogiannis, V.C.: Chess piece recognition using deep convolutional neural networks. In: 4th Symposium on Pattern Recognition and Applications (SPRA). vol. 13162, p. 1316202 (2024)
21. Papadimitriou, O., Kanavos, A., Mylonas, P., Maragoudakis, M.: Advancing weather image classification using deep convolutional neural networks. In: 18th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP). pp. 1–6. IEEE (2023)
22. Papadimitriou, O., Kanavos, A., Mylonas, P., Maragoudakis, M.: Classification of alzheimer's disease subjects from MRI using deep convolutional neural networks. In: 3rd International Conference on Novel & Intelligent Digital Systems (NiDS). Lecture Notes in Networks and Systems, vol. 784, pp. 277–286. Springer (2023)
23. Powers, D.M.W.: Evaluation: From precision, recall and f-factor to roc, informedness, markedness and correlation. CoRR **abs/2010.16061** (2020)
24. Ravoor, P.C., Sudarshan, T.S.B.: Deep learning methods for multi-species animal re-identification and tracking - a survey. Computer Science Review **38**, 100289 (2020)
25. dos Santos, C.F.G., Papa, J.P.: Avoiding overfitting: A survey on regularization methods for convolutional neural networks. ACM Computing Surveys **54**(10s), 213:1–213:25 (2022)
26. Savvopoulos, A., Kanavos, A., Mylonas, P., Sioutas, S.: LSTM accelerator for convolutional object identification. Algorithms **11**(10), 157 (2018)
27. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**, 60 (2019)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations (ICLR) (2015)
29. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
30. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: 36th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 97, pp. 6105–6114. PMLR (2019)
31. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6848–6856. Computer Vision Foundation / IEEE Computer Society (2018)