Comparative Analysis of Classic ML and Deep Learning Approaches for Bird Species Classification

Sujan Khanal
Department of Science and technology
University of Canberra
Canberra, Australia
u3258630@uni.canberra.edu.au

Abstract— **This project implemented and evaluated automated bird species recognition techniques using the Caltech-UCSD Birds-200-2011 dataset, containing images from 200 bird classes. Two distinct approaches were explored: classic machine learning pipelines utilizing handcrafted features, and deep learning-based models developed from scratch. In the classic approach, Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) features were extracted and classified using Support Vector Machines (SVM) and Random Forest classifiers. For deep learning, custom convolutional neural network (CNN) architectures were designed to automatically learn features and perform classification. Experiments were conducted using both the whole image and bounding box-cropped regions to evaluate model performance. Initial experiments on a 20-class subset allowed hyperparameter optimization and model refinement, after which the best models were applied to the full 200-class dataset. On the 200 Class, the custom CNN model achieved the highest accuracy of 33.43% on the bounding box-cropped images. Further applying fivefold cross-validation with the CNN model yielded an average accuracy of 41.45%, demonstrating consistent improvements. While transfer learning with ResNet-50 achieved a higher accuracy of 75.55%, it was included only for comparative purposes, with the primary focus remaining on the performance of the custom-built CNN and classic machine learning pipelines for fine-grained bird species recognition.**

Keywords— Bird Species Recognition; Image Classification Classic Machine Learning; SIFT; HOG; SVM; Random Forest; Deep Learning; CNN.

I. INTRODUCTION

Automated bird species recognition has gained significant attention in the fields of computer vision, biodiversity monitoring, and ecological research. Accurately identifying bird species from images remains a challenging task due to fine-grained differences between species, variations in pose, lighting conditions, and background clutter. Traditional approaches often rely on handcrafted feature extraction techniques combined with classical machine learning classifiers to address these challenges. More recently, deep learning methods, particularly convolutional neural networks (CNNs), have demonstrated the ability to learn hierarchical feature representations directly from raw image data, leading to significant improvements in image classification tasks.

This project investigated two primary approaches for bird species recognition using the Caltech-UCSD Birds-200-2011 dataset, which contains images from 200 different bird classes. The first approach employed handcrafted features such as Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG), classified using Support Vector Machines (SVM) and Random Forests (RF). The second approach involved developing custom CNN architectures to perform end-to-end feature learning and classification. A subset of 20 bird classes was initially used to conduct extensive experiments and hyperparameter optimization. The best-performing models were subsequently evaluated on the full 200-class dataset.

Experiments were conducted separately using the entire image and bounding box-cropped regions containing the birds to assess the impact of localization on classification performance. While transfer learning with ResNet-50 was briefly utilized for comparative purposes, the primary focus remained on evaluating the effectiveness of custom-designed CNN models and traditional machine learning pipelines.

II. METHODOLOGY

A. Dataset Description

The Caltech-UCSD Birds-200-2011 dataset was used for this project, comprising 11,788 images spanning 200 different bird species. Each image includes a corresponding bounding box annotation indicating the location of the bird. The dataset presents significant challenges due to subtle inter-class variations, variations in pose, and complex backgrounds.

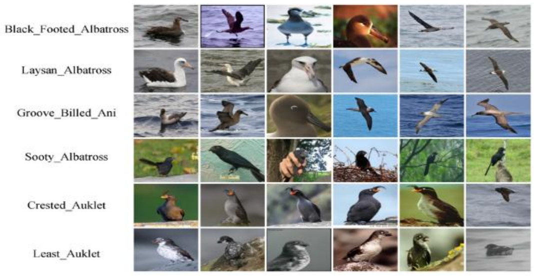

Fig. 1. Images-sampled-from-the-Caltech-UCSD-Birds-200-2011-CUB-dataset

B. Data Preparation

Two data preparation strategies were applied: using the entire image as input and using only the cropped region corresponding to the bounding box. For both scenarios, images were resized to standardized dimensions to ensure consistency across experiments. The dataset was partitioned into training (60%), validation (20%), and testing (20%) sets based on provided partition files, ensuring a balanced distribution across classes.

C. Feature Extraction and Classification Approaches

1. **Using Classic Machine Learning Algorithms:** Handcrafted feature extraction methods were first employed to represent the images by capturing important structural and texture-based information. These features were designed to highlight edges, key-points, and gradient patterns that are critical for distinguishing between different bird species.

Feature Descriptors:

- SIFT (Scale-Invariant Feature Transform): SIFT is a feature detection and description algorithm used in computer vision to identify and extract distinctive, scale- and rotation-invariant key-points from an image. It captures local image structures, such as corners or edges, and represents them using 128-dimensional feature vectors, making it effective to scale, rotation, and illumination changes.

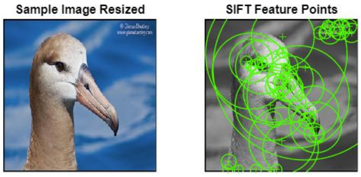

Fig. 2. SIFT Features

- HOG (Histogram of Oriented Gradients): HOG is a feature descriptor that represents the distribution of gradient orientations within localized portions of an image. It divides the image into small connected regions called cells, computes gradient directions or edge orientations for each pixel, and compiles them into histograms. HOG is particularly effective in capturing object shape and appearance, making it suitable for object detection tasks.

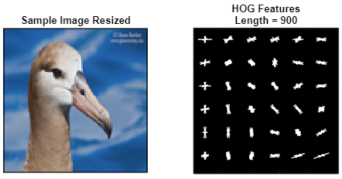

Fig. 3. HOG Features

Classic Machine Learning Algorithms:

- SVM (Support Vector Machine): SVM is a supervised machine learning algorithm used for classification and regression tasks. It aims to find the optimal hyperplane that maximally separates data points of different classes in a high-dimensional space. By using kernel functions (such as linear, polynomial, or RBF), SVM can handle both linear and non-linear classification problems effectively.
- Random Forest: Random Forest is an ensemble learning method that constructs a collection of decision trees during training and outputs the mode of the class predictions for classification tasks. It introduces randomness through bootstrapped sampling of the data and random selection of features at each split, which enhances model generalization and reduces overfitting.

Combined Approach ML:

- SIFT + SVM: - SIFT features were extracted from resized images and classified using SVM with radial basis function (RBF) kernels. Hyperparameters such as kernel scale and box constraint were optimized using Bayesian optimization.
- HOG + SVM: Histogram of Oriented Gradients (HOG) features were computed with varying cell sizes, and classification was performed using SVM. Hyperparameter tuning was similarly conducted.
- SIFT + Random Forest: SIFT features were classified using Random Forest classifiers. The number of trees and minimum leaf size were tuned to optimize performance.
- HOG + Random Forest: HOG features were used as input to Random Forest classifiers, with similar optimization strategies.

These approaches were evaluated separately on the whole image and bounding box-cropped image inputs.

2. **Deep Learning Approach:** CNN is a type of deep learning model particularly well-suited for processing grid-like data such as images. It automatically learns hierarchical feature representations through layers of convolution, activation functions (e.g., ReLU), pooling, and fully connected layers. CNNs are widely used for image classification, object detection, and other visual recognition tasks due to their ability to learn spatial hierarchies and reduce manual feature engineering.

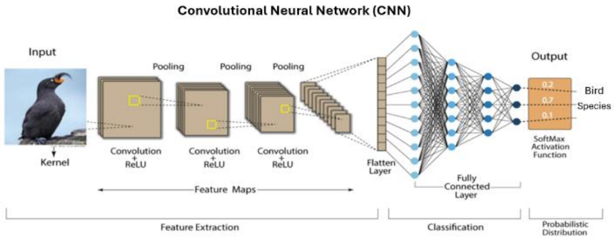

Fig. 4. CNN Algorithm

In this experiments, Custom convolutional neural network (CNN) architectures were designed to automatically learn features and perform classification. The networks consisted of multiple convolutional blocks, each containing convolutional layers, batch normalization, ReLU activation, dropout layers, and max-pooling. The models were trained using the Adam optimizer, with hyperparameters such as initial learning rate, mini-batch size, and number of epochs determined through empirical tuning.

While transfer learning using a pre-trained ResNet-50 was briefly explored for comparative purposes, the primary focus remained on evaluating the performance of the custom-built CNN models.

D. *Fivefold Cross-Validation*

For the fifth experiment, fivefold cross-validation was performed to further validate the effectiveness of the CNN model trained on bounding box-cropped images. Each class was partitioned into five subsets, and training, validation, and testing were rotated across these subsets according to the assignment guidelines. Average overall accuracy and average class-wise recognition rates were computed across the five runs to assess the model's generalization capabilities.

III. EXPERIMENTS AND DISCUSSIONS

The following listed experiments were conducted throughput this project:

	Classic handcrafted feature + classic ML classifier	Deep learning feature and Classifier
Whole Image as Input	Experiment 1	Experiment 2
Bounding Box as Input	Experiment 3	Experiment 4
5-fold Cross-Validation		Experiment 5

1. Subset 20 Classes: UCSD Birds-200-2011 dataset

TABLE I. EXP. 1: WHOLE IMAGE: CLASSIC-ML (20 CLASS BEST)

Features + ML Classifier	Handcraft Features Parameter	ML-Parameters	Accuracy (%)

SIFT + SVM
Target Size
[256 x 256]
	numFeatures = 20;
	maxFeatures = 400;
		KernelFunction = "rbf"
		Solver = 'SMO'
		Standarize = "true"
		Verbose='1'
			16.2

HOG + SVM
Target Size
[100 x 100]
	cellSize = [16 16]	KernelFunction = "rbf"
		Solver = 'SMO'
			21.2

SIFT +
Random Forest
numTrees=300
[256 x 256]
	numFeatures = 20;
	maxFeatures = 200;
		Method = 'Classification'
		OBBPrediction = 'ON'
			12.16

HOG +
Random Forest
numTrees=300
[100 x 100]
	cellSize = [16 16];	Method = 'Classification'
		OBBPrediction = 'ON'
		MinLeafSize = 5
		Prior = 'empirical'
			19.82

This experiment applied handcrafted feature extraction techniques—SIFT and HOG—combined with SVM and Random Forest classifiers, using resized full images for consistency. HOG + SVM achieved the best performance with 21.2% accuracy, aided by a [16×16] cell size that balanced locality and generalization, and an RBF kernel that captured non-linear decision boundaries. Resizing images to 100×100 preserved discriminative patterns while reducing complexity. SIFT + SVM achieved 16.2%, influenced by the number of detected keypoints (numFeatures = 20, maxFeatures = 400) and feature standardization. HOG + Random Forest achieved 19.82%, benefiting from 300 trees and a MinLeafSize of 5 to prevent overfitting. In contrast, SIFT + Random Forest performed worst at 12.16%, due to the incompatibility of sparse SIFT descriptors with tree-based ensemble methods.

TABLE II. EXP. 2: WHOLE IMAGE: DEEP-LEARNING(20 CLASS BEST)

Deep Learning Feature + Classifier	Layers	Hyperparameter	Param Values	Accuracy (%)
CNN [4 Blocks] [16-32-64-128]	InputLayer = [224, 224, 3] Normalised; Convolution 2dLayer; reluLayer; dropoutLayer =0.1; maxPooling2 dLayer	Solver	sgdm	**33.78**
		InitialLearnRate	0.001	
		MiniBatchSize	20	
		MaxEpochs	**15**	
		L2Regularization	None	
CNN [6 Blocks] [16:32:64:128:256:512] Target Size [128x128]	InputLayer = [128, 128, 3] Normalised; Convolution 2dLayer; reluLayer; maxPooling2 dLayer	Solver	adam	**51.80**
		InitialLearnRate	0.0005	
		MiniBatchSize	64	
		MaxEpochs	**30**	
		L2Regularization	None	
ResNet-50 Transfer Learning Target Size [224x224]	Pre-trained ResNet50 FullyConnect edLayer20 maxPooling2 dLayer	Solver	sgdm	**79.28**
		InitialLearnRate	0.0001	
		MiniBatchSize	16	
		MaxEpochs	**20**	
		L2Regularization	None	

In Experiment 2, a custom convolutional neural network (CNN) was trained on full images for end-to-end bird species classification, achieving 51.80% accuracy on the 20-class subset and outperforming all classic machine learning models. The CNN, consisting of 6 convolutional blocks with filter depths increasing from 16 to 512, automatically learned both low-level edges and high-level bird-specific features. Images were resized to [128×128×3] to balance detail and efficiency. Key training parameters included the Adam optimizer (initial learning rate 0.0005), mini-batch size 64, 30 epochs, and a dropout rate of 0.05 to improve generalization.

For comparison, a ResNet-50 model pre-trained on ImageNet was fine-tuned, achieving 79.28% accuracy with inputs resized to [224×224×3]. It was trained using the SGDM optimizer (learning rate 0.0001, batch size 16, 20 epochs). Although ResNet-50 delivered higher performance, it was included only for benchmarking, while the primary focus remained on evaluating handcrafted and custom CNN models.

TABLE III. EXP. 3: BOUNDING BOX: CLASSIC-ML(20 CLASS BEST)

Classic Handcraft Feature + ML Classifier	**Handcraft Features Parameter**	**ML-Parameters**	**Accuracy (%)**

HOG + SVM
Target Size
[100x100]
	cellSize = [16 16]	KernelFunction = "gaussian"
		KernalScale = 'auto'
		BoxConstraint = '998.46'
		Solver = 'SMO'
		Standarize = "true"
			29.3

HOG + Random Forest
numTrees=300
[100x100]
	cellSize = [16 16]	Method = 'Classification'
		OBBPrediction = 'ON'
		MinLeafSize = 5
		Prior = 'empirical'
			19.82

In Experiment 3, the same classic machine learning models and hyperparameters from Experiment 1 were applied, but using bounding box-cropped images instead of full images. Bounding box cropping led to a noticeable improvement in performance across all models by removing background clutter and forcing classifiers to focus on discriminative features of the birds. HOG + SVM achieved the best result with an accuracy of 29.3%, an increase from 21.2% in the whole image setting. This improvement confirms that localization enhances handcrafted feature effectiveness, although classic models still struggled with the fine-grained complexity compared to deep learning approaches.

TABLE IV. EXP. 4: BOUNDING BOX: DEEPLEARNING (20 CLASS BEST)

Deep Learning Feature + Classifier	Layers	Hyperparameter	Param Values	Accuracy (%)

CNN
[6 Blocks]
[16:32:64:128:256:512]
 	
Target-Size
[128x128]
	InputLayer = [128, 128, 3] Normalised;
	Convolution2dLayer;
	reluLayer;
	maxPooling2dLayer
		Solver	adam	**65.32**

		InitialLearnRate	0.0005	
		MiniBatchSize	64	
		MaxEpochs	**30**	
		L2Regularization	None	
ResNet-50 Transfer Learning	Pre-trained ResNet50 FullyConnectedLayer20 maxPooling2dLayer	Solver	sgdm	**83.33**
		InitialLearnRate	0.0001	
		MiniBatchSize	16	
		MaxEpochs	**20**	
		L2Regularization	None	

In Experiment 4, the custom CNN model and training settings from Experiment 2 were retrained using bounding box-cropped images instead of full images. Cropping the images to focus only on the bird regions led to a clear improvement in performance, with the CNN reaching an accuracy of 65.32%, compared to 51.80% when trained on full images. Removing background clutter helped the network focus better on important features such as feather patterns, beak shapes, and body structures, leading to stronger feature learning and better generalization. This result shows how important localized input is when dealing with fine-grained classification tasks in deep learning.

For comparison, a pre-trained ResNet-50 model was also fine-tuned on bounding box-cropped images. It achieved a higher accuracy of 83.33%, benefiting from deep feature representations learned from large datasets like ImageNet. While ResNet-50 outperformed the custom CNN, it was only included for benchmarking, and the main focus of the project remained on evaluating models built and trained from scratch.

2. Whole Image - 200 Classes: UCSD Birds-200-2011 dataset

TABLE V. Experiment 1: Whole Image: HOG + SVM (200 Class)

Classic Handcraft Feature + ML Classifier	Handcraft Features Parameter	ML-Parameters	Accuracy (%)

HOG + SVM
Target Size [100x100]
	cellSize = [16 16]	KernelFunction = “gaussian”
		KernalScale = ‘auto’
		BoxConstraint = ‘998.46’
		Solver = ‘SMO’
		Standarize = “true”
			6.1

Using HOG features and an RBF kernel SVM classifier on whole images yielded 6.1% accuracy. This performance drop from the 20-class subset reflects the model’s inability to scale in fine-grained, high-class-count settings. The model struggled due to background noise, poor localization, and limited feature abstraction. HOG captures edge orientation but lacks capacity for learning subtle patterns critical in distinguishing visually similar bird species. Background clutter further blurred discriminative features, and SVM’s linear boundaries proved insufficient across 200 complex classes.

TABLE VI. Experiment 2: Whole Image- CNN (200 Class)

Deep Learning Feature + Classifier	Layers	Hyperparameter	Accuracy (%)

CNN
[6 Blocks]
Target Size [128x128]
	InputLayer = [128, 128, 3] Normalised;
	Convolution2dLayer;
	reluLayer;
	maxPooling2dLayer
		Solver =adam
		InitialLearnRate =0.001
		MiniBatchSize =64
		MaxEpochs =30
		L2Regularization =NA
			21.53

A custom 6-block CNN trained on full images achieved 21.53% accuracy on the 200-class dataset. Despite not using localization, the CNN outperformed classic models due to its ability to learn hierarchical features directly from data. Inputs were resized to 128×128, and the model used Adam optimizer, batch size 64, and 30 epochs. Dropout and normalization improved generalization. Still, the presence of irrelevant background limited further performance. The CNN implicitly learned some object-focused features, but without explicit cropping, it often incorporated misleading background patterns during training.

TABLE VII. Exp. 3: Bounding Box- HOG+SVM (200 Class)

Classic Handcraft Feature + ML Classifier	Handcraft Features Parameter	ML-Parameters	Accuracy (%)

HOG + SVM
Target Size [100x100]
	cellSize = [16 16]	KernelFunction = “gaussian”
		KernalScale = ‘auto’
		BoxConstraint = ‘998.46’
		Solver = ‘SMO’
		Standarize = “true”
			16.2

With the same hyperparameters as Experiment 1, using bounding box-cropped images improved performance to 16.2%. Cropping removed background noise and allowed the model to focus solely on the bird region. HOG features became more relevant with localized input, capturing shape and texture better without interference. The RBF SVM benefitted from more consistent feature spaces across classes. However, classic features still lacked the richness and depth needed for modeling complex visual subtleties among 200 bird species, making this approach insufficient compared to deep learning.

TABLE VIII. Experiment 4: Whole Image- CNN (200 Class)

Deep Learning Feature + Classifier	Layers	Hyperparameter	Accuracy (%)

CNN
[6 Blocks]
Target Size [128x128]
	InputLayer = [128, 128, 3] Normalised;
	Convolution2dLayer;
	reluLayer;
	maxPooling2dLayer
		Solver =adam
		InitialLearnRate =0.001
		MiniBatchSize =64
		MaxEpochs =30
		L2Regularization=NA
			33.43

The custom CNN model trained on bounding box-cropped images achieved 33.43% accuracy, improving from 21.53% on whole images. Using localized input helped the model focus solely on bird features, removing irrelevant background information. The CNN, with 6 convolutional blocks and input resized to 128×128, leveraged dropout and batch normalization for regularization and stability. This setup enhanced the model’s ability to learn subtle class differences, such as variations in head markings or wing shapes. While still outperformed by deeper architectures, this result validated the effectiveness of end-to-end CNNs with targeted input for large-scale fine-grained classification.

TABLE IX. Experiment 4: Whole Image- ResNet-50 (200 Class)

Deep Learning Feature + Classifier	Layers	Hyperparameter	Accuracy (%)

ResNet-50 Transfer Learning
Target Size [224x224]
	Pre-trained ResNet50
	FullyConnectedLayer 20
	maxPooling2dLayer
		Solver =sgdm
		InitialLearnRate =0.0001
		MiniBatchSize =16
		MaxEpochs =20
		L2Regularization =NA
			75.55

The fine-tuned ResNet-50 model using bounding box-cropped images achieved 75.55% accuracy, the highest among all models. Leveraging pre-trained weights from ImageNet, ResNet-50 extracted highly generalizable visual features and benefited from deep residual connections that eased optimization. The model was retrained by replacing the final fully connected layer with a 200-class output layer. With inputs resized to 224×224, and hyperparameters including SGDM optimizer, batch size 16, and 20 epochs, ResNet-50 demonstrated the strength of transfer learning. However, it was used for benchmarking only, as the primary focus of this project was on custom CNN and classic ML models.

TABLE X. *Experiment 5: 5-folds Cross-Validation (200 Class)*

Deep Learning	Layers	Hyperparameter	Fold	Accuracy (%)

Feature + Classifier				
CNN [6 Block] TargetSize [128x128]	InputLayer = [128, 128, 3] Normalised; Convolution2dLayer; reluLayer; maxPooling2dLayer	Solver =adam	1-F	41.50
		InitialLearnRate =0.0005	2-F	43.11
		MiniBatchSize =64	3-F	43.37
		MaxEpochs =30	4-F	41.40
		L2Regularization=NA	5-F	37.86

To assess the generalization ability of the CNN model trained on bounding box-cropped images, a fivefold cross-validation strategy was applied to the 200-class dataset. Each class's images were split into five parts (20% each). In each run, three partitions were used for training, one for validation, and one for testing, maintaining the 60:20:20 ratio. This rotation was repeated across five runs to ensure comprehensive evaluation.

The model architecture remained the same as in Experiment 4. Cropped inputs helped the model focus on class-specific regions like wings, beaks, and plumage, essential in fine-grained classification.

The class-weighted overall average accuracy achieved by the CNN model across five cross-validation folds was 41.45%, reflecting consistent performance across all 200 bird species while accounting for class distribution. The model demonstrated strong generalization from cropped bird regions, even in a fine-grained classification setting. Analysis of correct and incorrect recognition per class showed good performance on visually distinct species like waterfowl and brightly coloured birds, but frequent misclassifications among similar species such as sparrows and finches. Some classes had high precision but low recall, indicating difficulty in fully capturing subtle distinguishing features.

In terms of stability and effectiveness, accuracy varied modestly (±2.5%) across folds. Fold 5 showed slightly lower accuracy (37.86%) due to a challenging test set with underrepresented classes. Overall, the model maintained reliable performance across different data partitions.

IV. CONCLUSIONS

This project demonstrated that deep learning approaches, particularly custom-designed CNNs, are more effective than classic machine learning pipelines for fine-grained bird species classification. The experiments showed that object localization through bounding box cropping significantly improves model focus and feature learning by reducing background noise. Furthermore, cross-validation confirmed the stability and effectives of the deep learning model across varying data partitions. Traditional handcrafted features, while simple and interpretable, proved insufficient for capturing subtle visual distinctions required for large-scale fine-grained recognition. Overall, the findings highlight the critical role of targeted input preprocessing and end-to-end feature learning in advancing performance in complex visual classification tasks.

V. RECOMMENDATIONS AND LESSON LEARNED

Through this project, it became evident that deep learning models, specifically convolutional neural networks (CNNs), are significantly more capable than classic machine learning pipelines in addressing fine-grained image classification tasks. The experiments highlighted the importance of object-focused inputs, as cropping images to bounding box regions led to noticeable performance improvements across all models. It was also observed that handcrafted features such as SIFT and HOG, while effective in simpler tasks, fail to capture the complexity required for distinguishing subtle inter-class variations among visually similar categories. Moreover, conducting fivefold cross-validation reinforced the necessity of robust evaluation practices to ensure that reported model performance is generalizable and not overly dependent on a specific data split.

Future work should consider the integration of data augmentation techniques such as random cropping, flipping, and colour jittering to further enhance model generalization and reduce overfitting. Employing part-based CNN architectures or attention mechanisms could help the network focus on the most discriminative parts of the birds, addressing issues where subtle differences drive class distinctions. Fine-tuning deeper pre-trained architectures beyond ResNet-50, such as DenseNet or EfficientNet, may yield additional performance gains. Finally, expanding the training dataset with additional annotated samples or synthetic images could mitigate class imbalance issues, particularly for underrepresented bird species, and enhance overall model efficiency.

VI. REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 2005, pp. 886–893.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems (NIPS), vol. 25, pp. 1097–1105, 2012.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.

[5] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA, USA: MIT Press, 2016.

[6] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in Proc. European Conf. Computer Vision (ECCV), Zurich, Switzerland, 2014, pp. 834–849.