Impact of Image Preprocessing Techniques on Machine Learning Models for Image Classification: A Case Study with Pokémon Images

Shreshth Kharbanda
Pokemon Classifier, Group #12
University of Washington
Seattle, WA
skhar@uw.edu

Siddanth Varanasi
Pokemon Classifier, Group #12
University of Washington
Seattle, WA
sidv2003@uw.edu

Ameya Agrawal
Pokemon Classifier, Group #12
University of Washington
Seattle, WA
aameya16@uw.edu

Aryan Mahindra
Pokemon Classifier, Group #12
University of Washington
Seattle, WA
armahin@uw.edu

Abstract

This study investigates the impact of various image preprocessing techniques and machine learning (ML) and deep learning (DL) models on the accuracy and computational efficiency of image classification in Augmented Reality (AR) and Virtual Reality (VR) gaming, specifically through the lens of Pokémon classification. Given the unique interactive requirements of AR/VR gaming, where real-time responsiveness and high performance are crucial, this research aims to identify the most effective methodologies for recognizing Pokémon characters under diverse environmental conditions. By employing and evaluating a range of preprocessing methods (including blurring, sharpening, and edge detection) on several ML/DL models (such as Convolutional Neural Networks, Dense Neural Networks, and traditional machine learning classifiers), this paper provides insights into optimizing image classification processes to enhance the AR/VR gaming experience. The results offer guidance on image classification in AR/VR games through the adoption of tailored preprocessing and modeling strategies. This research contributes to the broader field of computer vision and image classification, suggesting ways to optimize and advance interactive digital experiences across various applications.

1 Introduction

Advancements in AR and VR have led to the emergence of gaming experiences that blend digital content seamlessly with the physical world. These technologies enable games to interact with the environment in real-time, creating immersive user experiences. A key component of AR/VR gaming is the ability to accurately and quickly recognize various elements within the game, such as characters and objects, through image classification, which directly impacts the fluidity of user engagement and the user experience.

Given the complexity of real-world environments and the diverse conditions under which games are played, image classification in AR/VR presents unique challenges. These include but are not limited to variations in lighting, obfuscations, and the need to distinguish foreground from background. Pokémon games, particularly those leveraging AR technologies, exemplify these challenges due to their wide array of characters and the necessity for their recognition across different environments.

We evaluate the effectiveness of various image preprocessing techniques in different models to improve the accuracy and computational efficiency of Pokémon image classification. This research seeks to enhance the interactive gameplay experience in AR/VR games ensuring high performant and accurate classification models.

1.1 Individual Contributions

Member	**Contributions**
Shreshth	Data collection, verification and clean up; Implementation of model training (code)
Siddanth	Evaluation of models and graphing (code); Report write-up
Ameya	Research, selection and implementation of pre-processing techniques (code); Report write-up
Aryan	Implementation of preprocessing techniques (code); Presentation

2 Related work

2.1 Analysis of Convolutional Neural Networks for Document Image Classification

Abstract: Convolutional Neural Networks (CNNs) are state-of-the-art models for document image classification tasks. However, many of these approaches rely on parameters and architectures designed for classifying natural images, which differ from document images. We question whether this is appropriate and conduct a large empirical study to find what aspects of CNNs most affect performance on document images. Among other results, we exceed the state-of-the-art on the RVL-CDIP dataset by using shear transform data augmentation and an architecture designed for a larger input image. Additionally, we analyze the learned features and find evidence that CNNs trained on RVL-CDIP learn region-specific layout features.

2.2 A survey of image classification methods and techniques for improving classification performance

Abstract: Convolutional Neural Networks (CNNs) are state-of-the-art models for document image classification tasks. However, many of these approaches rely on parameters and architectures designed for classifying natural images, which differ from document images. We question whether this is appropriate and conduct a large empirical study to find what aspects of CNNs most affect performance on document images. Among other results, we exceed the state-of-the-art on the RVL-CDIP dataset by using shear transform data augmentation and an architecture designed for a larger input image. Additionally, we analyze the learned features and find evidence that CNNs trained on RVL-CDIP learn region-specific layout features.

2.3 Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review

Abstract: Convolutional neural networks (CNNs) have been applied to visual tasks since the late 1980s. However, despite a few scattered applications, they were dormant until the mid-2000s when developments in computing power and the advent of large amounts of labeled data, supplemented by improved algorithms, contributed to their advancement and brought them to the forefront of a neural network renaissance that has seen rapid progression since 2012. In this review, which focuses on the application of CNNs to image classification tasks, we cover their development, from their predecessors up to recent state-of-the-art deep learning systems. Along the way, we analyze (1) their early successes, (2) their role in the deep learning renaissance, (3) selected symbolic works that have contributed to their recent popularity, and (4) several improvement attempts by reviewing contributions and challenges of over 300 publications. We also introduce some of their current trends and remaining challenges.

2.4 Simple convolutional neural network on image classification

Abstract: In recent years, deep learning has been used in image classification, object tracking, pose estimation, text detection and recognition, visual saliency detection, action recognition and scene labeling. Auto Encoder, sparse coding, Restricted Boltzmann Machine, Deep Belief Networks and Convolutional neural networks is commonly used models in deep learning. Among different type of models, Convolutional neural networks has been demonstrated high performance on image classification. In this paper we bulided a simple Convolutional neural network on image classification. This simple Convolutional neural network completed the image classification. Our experiments are based on benchmarking datasets minist [1] and cifar-10. On the basis of the Convolutional neural network, we also analyzed different methods of learning rate set and different optimization algorithm of solving the optimal parameters of the influence on image classification.

2.5 Pokepedia: Pokemon Image Classification Using Transfer Learning

Abstract: Identifying images of various objects, living creatures, food, etc., and classifying them using machine learning has become a common task in computer vision. Humans may not identify every object they see, here comes machine learning that eases the life of human beings by identifying the object for the human. Pokémon is a cartoon that is widely watched by the majority of the younger generation around the world. The aim of this work to predict and classify Pokémon images using pre-trained models. In the proposed work, seven pre-trained models namely MobileNetV2, EfficientNetB7, EfficientNetV2L, DenseNet201, ResNet101, VGG19 and VGG16 were utilised to classify ten Pokémon characters which includes Pikachu, Raichu, Charmander, Bulbasaur, Squirtle, Eevee, Piplup, Snorlax, Jigglypuff, and Psyduck. The performance of the pre-trained models were evaluated on a dataset collected from the internet. The ResNet101 pre-trained model produces the highest accuracy of 95.60% when compared with the other models.

3 Background

The integration of Augmented Reality (AR) and Virtual Reality (VR) technologies within interactive applications, particularly in gaming, has necessitated interpreting and manipulating real-world images in real-time. Central to these methods are image preprocessing techniques and ML/DL models, each contributing uniquely to the enhancement of AR/VR experiences. This section delves into the technicalities of these components, outlining their significance in Pokémon image classification for AR/VR gaming.

3.1 Augmented Reality (AR) and Virtual Reality (VR)

Augmented Reality (AR) and Virtual Reality (VR) are two sides of immersive technology, yet they offer distinctly different experiences.

VR immerses users entirely in a computer-generated environment, detaching them from the physical world. Through VR headsets, users experience simulated environments that can replicate real-world settings or conjure fantastical worlds. This technology has profound applications in training simulations (military, medical training), entertainment (gaming, virtual concerts), and therapy (treatment of PTSD), showcasing its ability to create fully controlled and immersive virtual experiences.

AR enhances the real world by overlaying digital information (text, images, 3D models) onto it. Users interact with both physical and virtual elements using devices like smartphones, tablets, or AR glasses. Applications of AR span various fields, including education (interactive learning tools), healthcare (surgical visualization), and retail (virtual try-ons), reflecting its versatility in enhancing real-world experiences with digital information.

AR/VR gaming represents a significant leap from traditional gaming, offering immersive experiences that blend physical and digital worlds. Pokémon GO, one of the most prominent examples of AR gaming, utilizes GPS and AR technologies to overlay Pokémon characters onto the real world, as seen through the player's device. This integration encourages physical exploration, making the gaming experience interactive and engaging.

3.2 Machine Learning and Deep Learning

3.2.1 Machine Learning (ML)

ML is a set of algorithms and statistical models that enable computers to perform specific tasks without explicit instructions, relying instead on patterns and inference. ML models, such as Support Vector Machines (SVMs), Random Forests, and Gradient Boosting Machines, have been widely used in image classification tasks. These models excel in handling structured data and can be very effective when combined with appropriate feature extraction techniques.

3.2.2 Deep Learning (DL)

DL, a subset of ML, is particularly suited to processing unstructured data like images. DL models, especially Convolutional Neural Networks (CNNs), have revolutionized image classification tasks because they can automatically extract features from raw images.

Convolutional Neural Networks (CNNs) CNNs are designed to automatically learn spatial hierarchies of features from images. CNN architecture includes convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification.

Dense Neural Networks (DNNs) DNNs involve multiple layers of interconnected neurons and can learn complex patterns from high-dimensional data. DNNs are often used in tasks like image classification, where the raw image data is flattened and processed through deep, dense neural layers.

3.2.3 Model Types and Their Applications

Random Forest Classifier An ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes of the individual trees. Random Forests are known for their effectiveness in classification tasks by capturing diverse patterns and reducing overfitting.

Gradient Boosting Classifier Boosting methods work by consecutively correcting the mistakes of weak learners to improve classification accuracy. Gradient Boosting builds an additive model resulting in a powerful classifier that can highlight image features.

Support Vector Machine (SVM) In image classification, SVMs can efficiently perform a non-linear classification by implicitly mapping inputs into high-dimensional feature spaces.

Logistic Regression Logistic regression can be extended to multi-class classification to predict multiple classes of Pokémon from images. It's useful for probabilistic frameworks where understanding the confidence level of predictions is important.

3.3 Image Preprocessing Techniques

Image preprocessing techniques play a pivotal role in image classification models for AR and VR gaming. These techniques improve the quality and the interpretability of input images, addressing issues such as noise, distortion, and variations that can detract from the performance of ML/DL models.

3.3.1 Blurring: Gaussian Filter

Gaussian blurring mitigates noise and reduces detail in images with a Gaussian filter. This method uses a Gaussian function to smooth the image by averaging the pixels' values within a local neighborhood defined by the Gaussian kernel. The degree of smoothing is controlled by the parameter sigma (σ), which determines the width of the Gaussian kernel. In the context of Pokémon character recognition, Gaussian blurring can help minimize the impact of background noise and texture variations, allowing the classification models to focus on more relevant features.

3.3.2 Sharpening: Unsharp Masking

Unsharp masking enhances the edges and contrast of images, making the features within the images more distinct. It operates by subtracting a blurred version from the original image, which emphasizes the edges. Parameters control the scale of the edges to be enhanced and the intensity. Applying unsharp masking in AR/VR gaming contexts, such as

Pokémon classification, can help outline characters from their backgrounds helping the model leverage contours and textures.

3.3.3 Edge Detection: Sobel and Prewitt Operators

Edge detection techniques identify the boundaries or edges within images. The Sobel and Prewitt operators are among the most common methods for edge detection, designed to highlight regions of high spatial gradient in an image that correspond to edges.

Sobel Operator The Sobel operator uses a pair of 3x3 convolutional kernels, one estimating the gradient in the horizontal and vertical direction. The gradient magnitude for each pixel is then computed, highlighting the edges in the image. This operator is particularly effective in detecting edges in areas with a gradual intensity variation.

Prewitt Operator Similarly, the Prewitt operator employs two 3x3 convolutional kernels for horizontal and vertical edge detection. However, the Prewitt kernels are designed to have different weightings which emphasizes the linear features such as specific postures or shapes. We thought this one would be interesting to try because it is another type of edge detection filter we found from Chandra.

3.3.4 High-Pass Filtering

High-pass filtering is used to retain the high-frequency components of an image and eliminate the rest. It substracts a blurred version of the image from the original image to emphasize edges and fine details. High-pass filtering can highlight the distinguishable features especially in complex scenes.

4 Methodology

4.1 Dataset Acquisition and Preparation

The dataset was found on Kaggle (https://www.kaggle.com/datasets/lantian773030/pokemonclassification) consisting of 7,000 hand-cropped images of first-generation Pokémon characters with a balanced representation of 150 unique species. The preparation involved several key steps:

4.1.1 Initial Inspection

Every image underwent a detailed review to identify and eliminate any instances of corruption. This process ensured reliable model training and validation.

4.1.2 Image Resizing and Normalization

To standardize input dimensions for the neural networks, all images were resized to 128x128 pixels. This ensured that the models are not biased by image size variations. Then, pixel values were normalized to a [0, 1] range, enhancing model training stability and efficiency.

4.1.3 Stratified Dataset Splitting

The dataset was divided into an 80% training set and a 20% validation set, employing stratified sampling to maintain an equitable distribution of Pokémon classes across both subsets.

4.2 Image Preprocessing

Image preprocessing is a pivotal phase designed to optimize the input images for better feature extraction by the models. The chosen preprocessing techniques include:

4.2.1 Gaussian Blurring

Implemented to soften image details and reduce noise, blurring is applied using a Gaussian function with a σ of 2. This specific σ value was selected to adequately blur the images. The blurring process aids in de-emphasizing background elements that might confuse the models during character recognition.

4.2.2 Unsharp Masking

This technique sharpens the images, enhancing the contrast at the edges. By specifying a radius of 1 and an amount of 1, the implementation strikes a balance between edge enhancement and avoiding hallucinated artifacts.

4.2.3 Edge Detection

Sobel and Prewitt operators, applied in horizontal and vertical directions, highlight the boundaries within images. These operators emphasizing silhouettes differentiating Pokémon characters from backgrounds.

4.2.4 High-Pass Filtering

By subtracting a low-frequency version of the image from the original, high-pass filtering emphasizing distinctive features to Pokémon characters. This assists the models in recognizing subtle differences between species.

4.3 Model Selection, Architecture, and Training

Training algorithms were developed to ensure exhaustive learning from the training data, incorporating preprocessing techniques via TensorFlow's `ImageDataGenerator`. This approach allowed for real-time image augmentation, exposing the models to various imaging conditions reflective of real-world scenarios.

A diverse array of ML and DL models was chosen to explore their relative strengths and weaknesses in classifying the Pokémon images:

4.3.1 Convolutional Neural Networks (CNNs)

CNN architectures were designed with layers strategically arranged to incrementally extract and condense relevant features from the images. Other architectural decisions, such as the number of filters in convolutional layers and the dimensionality of dense layers, were informed by the complex characteristics of the characters in the dataset.

4.3.2 Dense Neural Networks (DNNs)

The construction of DNN models aimed to assess the performance of fully connected architectures in extracting patterns from flattened image inputs. The layer configuration and neuron count were optimized based on experimental evaluations, focusing on maximizing model accuracy while preventing overfitting.

4.3.3 Traditional ML Models

The selection of Random Forest, Gradient Boosting, SVM, and Logistic Regression models provided a comparative baseline to the neural network approaches. These models were integrated with various preprocessing techniques that included feature extraction and optional dimensionality reduction, preparing the image data for efficient processing.

4.4 Technical Implementation & Evaluation Metrics

The technical stack for this research includes TensorFlow for constructing and training CNN and DNN models, and scikit-learn for implementing traditional ML models. Advanced image preprocessing was done through a combination of TensorFlow's image processing capabilities and scikit-image's advanced filtering functions. The deployment of GPU acceleration was strategic, aimed at managing the computational demands of processing a large dataset and complex model architectures efficiently.

The evaluation framework centers on two primary metrics:

4.4.1 Classification Accuracy

Measured on the validation set, accuracy serves as the principal indicator of model performance, reflecting the proportion of correctly classified Pokémon images.

4.4.2 Training Time

Recorded for each model, training time provides insight into the computational efficiency of the model training process, a crucial consideration for real-time AR/VR applications.

4.4.3 Algorithm Psuedocode

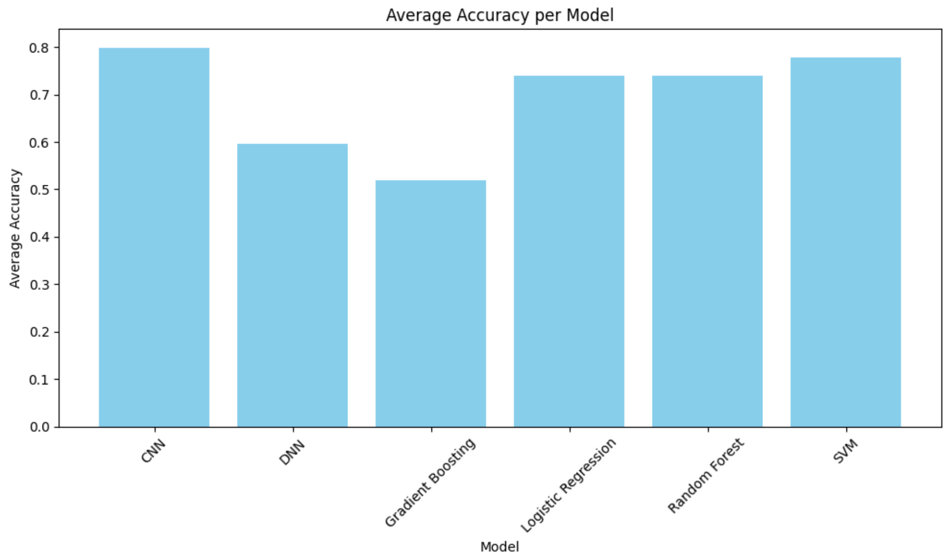
```
preprocessing_functions = {
    "None": None,
    "Blur": blur_image,
    "Sharpen": sharpen_image,
    "High-PassFilter": high_pass_filter,
    "SobelHorizontal": sobel_horizontal,
    "SobelVertical": sobel_vertical,
    "PrewittHorizontal": prewitt_horizontal,
    "PrewittVertical": prewitt_vertical
}

model_types = {
    "CNN": create_model,
    "DNN": create_dnn_model,
    "RandomForest": create_random_forest_model,
    "GradientBoosting": create_gradient_boosting_model,
    "SVM": create_svm_model,
    "LogisticRegression": create_logistic_regression_model
}

results = []

for model_name, model_creator in model_types.items():
    for preprocess_name, preprocess_function in preprocessing_functions.items():
        datagen = create_datagen(preprocessing_function=preprocess_function)
        train_generator = datagen.flow_from_directory(data_dir,
        target_size=(128, 128), batch_size=32,
        class_mode='categorical', subset='training')

        validation_generator = datagen.flow_from_directory(data_dir,
        target_size=(128, 128), batch_size=32,
        class_mode='categorical', subset='validation')

        model = model_creator(input_shape=(128, 128, 3))

        start_time = time.time()

        model.fit(train_generator, epochs=10,
        validation_data=validation_generator, verbose=1)

        training_time = time.time() - start_time

        _, accuracy = model.evaluate(validation_generator, verbose=0)

        results.append({
            "Model": model_name,
            "Preprocessing": preprocess_name,
            "Accuracy": accuracy,
            "TrainingTime": training_time
        })
```

5 Results

Table 1: Model Performance by Preprocessing Technique

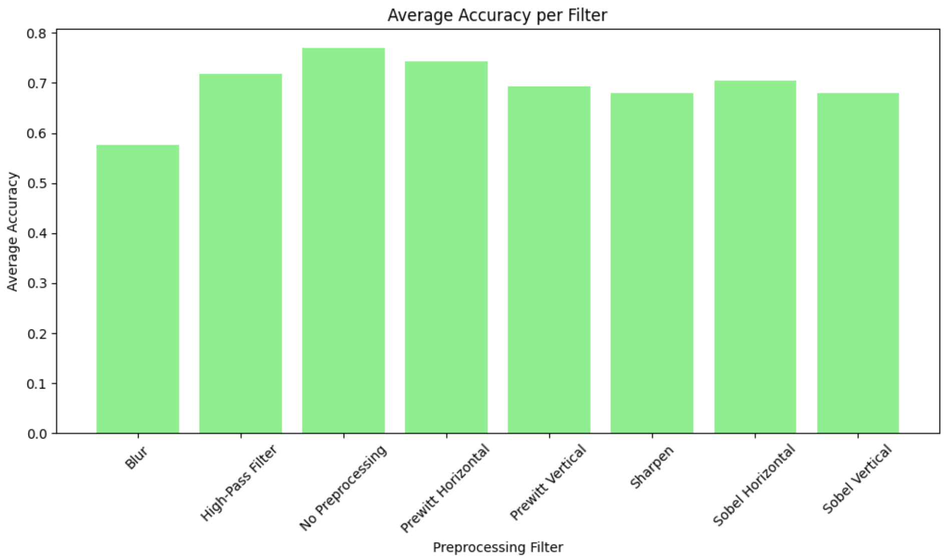
Model Type	Preprocessing	Accuracy	Training Time (s)
CNN	No Preprocessing	0.923	48.06
	Blur	0.615	56.63
	Sharpen	0.769	48.03
	High-Pass Filter	0.769	48.90
	Sobel Horizontal	0.769	63.24
	Sobel Vertical	0.769	51.69
	Prewitt Horizontal	0.846	56.05
	Prewitt Vertical	0.923	48.92
DNN	No Preprocessing	0.692	45.77
	Blur	0.462	44.35
	Sharpen	0.769	51.48
	High-Pass Filter	0.615	54.42
	Sobel Horizontal	0.692	52.09
	Sobel Vertical	0.385	49.00
	Prewitt Horizontal	0.615	48.37
	Prewitt Vertical	0.538	47.62
Random Forest	No Preprocessing	0.846	1.16
	Blur	0.692	2.41
	Sharpen	0.538	3.17
	High-Pass Filter	0.769	1.91
	Sobel Horizontal	0.769	1.14
	Sobel Vertical	0.769	1.00
	Prewitt Horizontal	0.769	1.21
	Prewitt Vertical	0.769	1.16
Gradient Boosting	No Preprocessing	0.538	16.17
	Blur	0.385	12.49
	Sharpen	0.462	6.07
	High-Pass Filter	0.615	23.35
	Sobel Horizontal	0.462	13.85
	Sobel Vertical	0.615	12.10
	Prewitt Horizontal	0.692	16.96
	Prewitt Vertical	0.385	17.40
SVM	No Preprocessing	0.846	1.29
	Blur	0.769	1.36
	Sharpen	0.769	1.02
	High-Pass Filter	0.769	1.61
	Sobel Horizontal	0.769	1.08
	Sobel Vertical	0.769	1.37
	Prewitt Horizontal	0.769	1.76
	Prewitt Vertical	0.769	1.51
Logistic Regression	No Preprocessing	0.769	110.75
	Blur	0.538	126.19
	Sharpen	0.769	4.22
	High-Pass Filter	0.769	10.67
	Sobel Horizontal	0.769	8.09
	Sobel Vertical	0.769	11.19
	Prewitt Horizontal	0.769	17.91
	Prewitt Vertical	0.769	8.52

Table 2: Model Architectures and Their Impact on Accuracy



Model	Accuracy
CNN	0.798
DNN	0.596
Gradient Boosting	0.519
Logistic Regression	0.740
Random Forest	0.740
SVM	0.779

Table 3: Preprocessing Techniques and Their Impact on Accuracy



Preprocessing	Accuracy
Blur	0.577
High-Pass Filter	0.718
No Preprocessing	0.769
Prewitt Horizontal	0.744
Prewitt Vertical	0.692
Sharpen	0.679
Sobel Horizontal	0.705
Sobel Vertical	0.679

6 Discussion

Table 1, Table 2, and Table 3 summarize our findings. The following synthesis draws from the above, and provides insights into what these results might mean.

6.1 Comparison of Model Performance:

- **CNNs** showed the highest accuracy when averaged over all preprocessing techniques (0.798) among all models, with no preprocessing and the Prewitt Vertical filter achieving the highest individual accuracies (0.923). This indicates that CNNs are particularly effective for image classification tasks, likely because of their ability to capture intricacies in complex images.
- **SVMs** also performed well, with an average accuracy of 0.779. Their consistent performance across all filters suggests that SVMs are consistently strong across different preprocessing techniques.
- **Random Forest** and **Logistic Regression** models exhibited similar average accuracies (0.740), which, while not as high as CNNs or SVMs, still demonstrates their effectiveness.
- **DNNs** and **Gradient Boosting** had the lowest average accuracies (0.596 and 0.519, respectively), suggesting that these models might require more specialized tuning or might not be as suitable for the specific task of Pokémon image classification without significant adjustments.

6.2 Comparison of Model Efficiency:

- CNN generally exhibited higher accuracies across different preprocessing techniques compared to other models, especially with Sobel and Prewitt vertical preprocessing, achieving accuracies ranging from 0.615 to 0.923. However, CNNs tend to have longer training times, particularly with Sobel horizontal preprocessing, taking around 63.24 seconds.

- Random Forest models demonstrate decent accuracies with relatively short training times, especially without preprocessing, achieving one of the best results for the real-time AR/VR gaming use case: an accuracy of 0.846 in just 1.16 seconds.
- DNN and Gradient Boosting models showcase some of the lowest accuracies. Gradient Boosting at least had reasonable training times, but DNNs were on the higher side of training time as well. The preprocessing techniques affect both aspects variably for both models, so there wasn't a conclusive pattern.
- SVM models perform consistently across preprocessing techniques in terms of accuracy and training time, maintaining a great balance.
- Logistic Regression models, while achieving competitive accuracies, tend to have longer training times compared to other models, particularly when No preprocessing and Blur preprocessing techniques are applied.

6.3 Comparison of Preprocessing Techniques:

- No Preprocessing led to the highest average accuracy (0.769), underlining the quality and distinctiveness of the original images for Pokémon classification. This also suggests that the inherent features of Pokémon images are already quite suitable for classification without the need for preprocessing.
- Among the preprocessing techniques, Blur had the lowest average accuracy (0.577), indicating that it may degrade important or distinguishing features in Pokémon images used for classification.
- Prewitt Horizontal and High-Pass Filter techniques showed improved accuracies (0.744 and 0.718, respectively), suggesting these methods might enhance relevant features or reduce noise, enhancing the classification process.

6.4 Strengths:

- **High Performance of CNNs:** One of the clear strengths identified in the study is the superior performance of Convolutional Neural Networks (CNNs) in classifying Pokémon images. This reinforces the notion that CNNs, with their ability to capture spatial and temporal dependencies, are particularly well-suited for image recognition tasks within AR/VR environments.
- **Effective Use of Certain Preprocessing Techniques:** The effectiveness of specific preprocessing techniques like Prewitt Horizontal and High-Pass Filter in improving classification accuracy points towards the potential of targeted preprocessing in enhancing the model's performance.

6.5 Weaknesses:

- **Underperformance of DNNs and Gradient Boosting:** The lower average accuracy of Deep Neural Networks (DNNs) and Gradient Boosting models highlights a weakness in the applicability of these models to the specific task of Pokémon image classification without significant customization or tuning.
- **Negative Impact of Some Preprocessing Techniques:** The application of the Blur preprocessing technique, which led to a decrease in accuracy across models, underscores a potential weakness in applying preprocessing indiscriminately. It suggests that not all preprocessing techniques are beneficial and their impact must be assessed in the context of the specific classification task.

6.6 What Worked:

- **No Preprocessing for CNNs:** The high accuracy achieved by CNNs without preprocessing or with minimal preprocessing suggests that the original quality of Pokémon images is sufficient for effective classification. This finding is significant as it indicates that, for certain datasets and models, preprocessing may be unnecessary and could even degrade performance.
- **Selective Preprocessing Improvements:** Preprocessing techniques that focus on enhancing specific features (e.g., edges) of images have been shown to improve accuracy for some models. This indicates that selective and targeted preprocessing can be beneficial.

6.7 What Didn't Work:

- **Blur Preprocessing Technique:** The application of the blur filter consistently resulted in lower accuracies. This suggests that blurring, which reduces image detail, might be counterproductive for distinguishing between different Pokémon characters, which often have small, distinctive features.

- **Generalized Approaches for DNNs and Gradient Boosting:** The approach of employing DNNs and Gradient Boosting without specific optimizations did not yield the expected levels of accuracy. This suggests that a more tailored approach, possibly involving architecture adjustments or hyperparameter tuning, might be necessary for these models to perform effectively in this domain.

7 Conclusion

7.1 Conclusions

In this study, we explored the accuracy of various machine learning models and pre-processing techniques for AR/VR applications through the lens of image classification of Pokémon images. This lens encompasses both entertainment technology and data analysis. We built various models and tried different filters with each model and our findings highlight the exceptional performance of CNNs. This follows their popularity for handling complex image recognition tasks, such as those commonly seen in AR/VR environments (largely in gaming environments). Furthermore, the investigation into pre-processing techniques revealed that while some filters like Prewitt Horizontal and High Pass Filter can enhance model accuracy, others like the Blur Filter, might detract from a model's classification ability. The underperformance of DNNs was unexpected as they are generally considered highly effective in image classification (as seen in the work by Rawat and Wang). A possible explanation for this is that these models need further customization, along with a deeper exploration of the ideal combination of filtering which could help achieve much higher accuracies for this particular task. This nuanced understanding of the impact of various models and preprocessing techniques offers valuable insights into the development of more effective AR/VR applications, where accurate and rapid image classification is extremely important.

7.2 Future Work

The outcomes of this study establish the basis for several avenues of future research, particularly in extending the scope of image classification techniques within AR/VR gaming environments beyond simply Pokémon. One avenue is exploring the integration of real-time environmental context awareness to refine the accuracy of Pokémon, or more broadly, character recognition under various conditions subject to change such as lighting, backgrounds (specifically in motion), weather, and even season. Another avenue involves extending the scope of this project to introduce more sophisticated pre-processing techniques such as adaptive filtering and accompanying this with newer and advanced methods such as deep-learning based segmentation methods. This can help gain a better understanding of the effectiveness of applying filters for image classification, along with potentially enhancing accuracy. Another avenue is the investigation of transfer learning to enable the models to rapidly adapt to new, unseen Pokémon or other characters with minimal training data. This is further supported by Ch et al., who explored the effect of transfer learning on Pokemon classification and achieved an accuracy of 95%, highlighting its effectiveness on image classification tasks. Finally, with how intense AR/VR applications may become in computing and energy, future work could also assess the efficiency of deploying such models on portable AR/VR devices to ensure stability and sustainability alongside technological advancement.

Ultimately, this research provides a strong basis for further analyzing the effects of various image classification techniques for character recognition. This, along with future efforts, will contribute to achieving seamless, highly responsive, and immersive experiences (especially gaming) in AR/VR platforms, enriching the user experience and propelling the AR/VR industry to newer heights.

References

[Ch et al., 2023] Ch, V., Syeda, A., Kolla, N., Ghanta, N., and Muvva, S. (2023). Pokepedia: Pokemon image classification using transfer learning. *Review of Computer Engineering Studies*, 10:14–19.

[Chandra, 2020] Chandra, B. (2020). A beginners guide to computer vision (part 2)- edge detection. `https://medium.com/analytics-vidhya/a-beginners-guide-to-computer-vision-part-2-edge-detection-4f10777d5483`.

[Guo et al., 2017] Guo, T., Dong, J., Li, H., and Gao, Y. (2017). Simple convolutional neural network on image classification. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pages 721–724.

[Rawat and Wang, 2017] Rawat, W. and Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449.

[Tensmeyer and Martinez, 2017] Tensmeyer, C. and Martinez, T. (2017). Analysis of convolutional neural networks for document image classification. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 388–393.

[Zhang, 2020] Zhang, L. (2020). 7,000 Labeled Pokemon. `https://www.kaggle.com/datasets/lantian773030/pokemonclassification`. Accessed: 2024-02-25.