Analyzing MLP and CNN on Pokémon Type Classification

Quenton Ni, Vy Bui Nguyen, Liem Tran

May 2025

1 Abstract

This project investigates deep learning models for classifying Pokémon by their elemental types using stylized images. Utilizing a dataset sourced from Kaggle, we evaluated two(2) multilayer perceptrons (MLPs) and two(2) convolutional neural networks (CNNs) on a multi-class image classification task. Although CNNs demonstrated strong training performance, they experienced significant overfitting and underperformed on unseen data. Conversely, MLPs exhibited better generalization despite their simpler architecture. This study highlights key challenges such as dataset imbalance and stylized image inconsistencies, proposing future improvements like enhanced regularization, class balancing, data augmentation, and transfer learning.

2 Introduction

Image classification is a fundamental task in machine learning and computer vision, significantly enhanced by recent advancements in deep learning. While most research traditionally focuses on natural images, there is increasing interest in applying these techniques to stylized or synthetic imagery. Stylized datasets such as hand-drawn art, cartoons, and game sprites present unique challenges for machine learning models, including exaggerated or inconsistent visual features and limited data availability. Exploring these domains pushes the boundaries of what current architectures can achieve, especially when trained on less standardized or lower resource inputs.

Pokémon, fictional creatures categorized into elemental "types" such as Fire, Water, Grass, or Electric, provide an intriguing domain for such exploration. These types are fundamental to gameplay and are generally recognizable through consistent visual traits like flames, aquatic textures, or leaf motifs. The diversity of Pokémon and the artistic liberties in their designs allow for a complex yet structured classification task, which is ideal for evaluating model performance in stylized environments.

In this project, we aim to classify Pokémon into their assigned elemental types based solely on stylized image data from a gamified dataset. By using models that operate without any textual or gameplay context, we test the extent to which visual features alone can drive accurate classification. We implement and evaluate multiple deep learning architectures, specifically, multilayer perceptrons (MLPs) and convolutional neural networks (CNNs), to compare their effectiveness on this task. Through this evaluation, we hope to gain insights into the adaptability of these architectures for unconventional visual inputs, with implications for broader applications in domains like gaming, animation, and other stylized content.

3 Background

3.1 Introduction

Image classification is a core task in computer vision, and deep learning has significantly improved its performance across a range of applications. Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) are commonly used models that have shown strong results in identifying and categorizing image content. This project explores the task of predicting the type(s) of a Pokémon based on its image, using image classification techniques. While Pokémon are fictional characters, this task poses real-world challenges such as handling stylized or synthetic images, modeling dual labels since many Pokémon have two types, and training with relatively limited data. The goal of this literature review is to examine a selection of peer-reviewed research papers related to image classification, multi-label learning, and domain-specific datasets. By reviewing methods and findings from recent work, we aim to identify effective techniques that can be applied to this project and understand the current limitations in the field.

We divide our literature review into five key areas relevant to our work:

- CNNs in image classification
- Multi-label learning
- Transformer-based architectures
- Transfer learning
- Domain-specific applications in stylized imagery

3.2 Convolutional Neural Networks (CNNs) for Image Classification

Dhruv and Naskar [3] review how combining CNNs and Recurrent Neural Networks (RNNs) can enhance image classification, especially for images with complex or layered information. While CNNs excel at extracting visual features, RNNs help models capture relationships

between those features. This blended approach is particularly relevant to our project, where different parts of a Pokémon's design may hint at multiple types. This hybrid architecture is highly relevant for our project, where different visual elements of a Pokémon, such as fire patterns, wings, or aquatic textures, may suggest multiple types simultaneously. Integrating CNN and RNN strengths could help our model reason beyond isolated features and capture more holistic type signatures. Chaganti et al. [2] compare Support Vector Machines (SVMs) and CNNs, showing that although CNNs are powerful due to their ability to learn features directly from data, SVMs can still perform well when dealing with simpler feature sets or smaller datasets. Given that our Pokémon dataset is both relatively small and artistically varied, these insights help us think carefully about balancing model complexity and performance. It may be wise to remain cautious of overfitting and consider simpler classifiers as baselines or ensemble components. Mishra and Bhavsar [11] take a different approach, studying how humans classify images by analyzing EEG brain signals. While outside the direct scope of our work, their research offers an interesting parallel: humans leverage contextual, intuitive cues when interpreting complex images, something most machine models still struggle to emulate. In our case, considering how humans intuitively recognize a Fire or Water type from subtle stylistic hints can inspire model designs that are sensitive to low-level and high-level features alike.

3.3 Multi-Label Classification Techniques

Handling multi-label classification effectively is crucial for our project, as many Pokémon have two types that are meaningfully related (e.g., Fire/Flying, Water/Ice). Integrating convolutional models with sequential architectures, as shown in the work by Wang et al. [15], offers a promising solution. Their architecture first extracts features using a CNN, then models label dependencies sequentially with an RNN. This structure is valuable for our project because Pokémon types are not independent, certain types often co-occur with others in logical patterns (e.g., Grass frequently pairing with Poison). Capturing these co-occurrence patterns could significantly enhance our model's predictive accuracy. Supporting this need for multi-label sensitivity, Endut et al. [4] further reinforces the importance of this design considerations. They show that both traditional and deep learning methods benefit from loss functions and activation strategies (such as sigmoid rather than softmax) that accommodate multiple, non-exclusive labels, a vital point for our dual-type classification task. Triguero and Vens [13] dive deeper into hierarchical multi-label classification, exploring how structuring labels (rather than treating them independently) can significantly improve performance. Their work suggests that organizing labels into meaningful hierarchies can improve performance by capturing latent structure among labels. Applying a hierarchy to Pokémon types, for instance, grouping Dragon, Flying, and Ground under a broader "elemental" category, could help our model learn associations more effectively, especially for rarer type combinations.

3.4 Transformer-Based Models (ViT and C-Tran)

Transformers offer new possibilities for image classification, especially for complex or layered visuals. Unlike CNNs, which process images through local receptive fields, transformers use self-attention mechanisms to model global dependencies across an entire image. Wang et al. [16] highlight how Vision Transformers (ViTs) differ from CNNs by using attention mechanisms that capture broader, more global patterns. In our case, where visual cues connecting to different Pokémon types might be subtle, this flexibility could greatly enhance model performance. More specifically, transformer-based models like C-Tran [9] provide a compelling framework for multi-label prediction. C-Tran's cross-attention mechanism allows different labels to attend to different regions of an image, meaning "Fire" can focus on flame-like visuals, while "Water" can focus on cooler tones. For our project, this label-specific attention is a perfect match for the nuanced designs of Pokémon. Similarly, Dong et al. [14] show the success of ViTs in multi-label medical imaging tasks, emphasizing the strength of self-attention mechanisms for capturing complex dependencies. Their work gives us confidence that ViTs and related architectures can handle the intricate visuals and label co-occurrences present in our Pokémon dataset.

3.5 Transfer Learning and Fine-Tuning

Given that our dataset is relatively small, transfer learning strategies are especially relevant. Plested and Gedeon [12] discuss how reusing models trained on large datasets like ImageNet allows researchers to adapt powerful feature extractors with much less data. This matches our need to efficiently build a high-performing model without requiring massive amounts of Pokémon training images. Leveraging pre-trained models, we can bootstrap strong visual feature extractors, reducing the need for extensive training from scratch. Alfano et al. [1] introduce an even more lightweight approach called Top-Tuning, which freezes the feature extractor and only trains a simple classifier on top. This strategy could help us quickly fine-tune a strong model without the computational burden of full retraining, making it an attractive option for our project timeline. Additionally, Gu, Pednekar, and Slater [5] emphasize the importance of data augmentation techniques, like rotation, scaling, and color adjustments, for improving model generalization, especially when datasets are small. For our Pokémon images, careful augmentation could help the model become more robust to style variations and artistic exaggerations. Furthermore, Kim et al. [6] show that transfer learning is especially effective in domains like medical imaging, where datasets are small and highly specialized. Their findings support our decision to leverage pre-trained models for classifying Pokémon types, which similarly feature stylized and domain-specific visual characteristics.

3.6 Domain-Specific Applications (Cartoon/Game Characters)

Working with fictional or stylized imagery presents challenges distinct from real-world datasets. The Open Images Dataset V4 [8] provides an example of a massive, richly annotated dataset

supporting tasks like multi-label classification and visual relationship recognition. Although it focuses on real-world images, the design principles behind Open Images, rich labeling, diverse object representations, and attention to overlapping categories inform our approach to working with Pokémon. Further addressing synthetic and stylized data, Kaur et al. [7] review synthetic image generation techniques and highlight challenges like exaggerated features and style inconsistencies. These insights are highly relevant for our task, given that Pokémon artwork often features artistic exaggerations rather than photorealistic details. Specific to animated or fictional characters, Lyu, Lee, and Liu [10] propose the LMV-ACGAN model, a hybrid CNN-GAN architecture that enhances color fidelity and feature extraction for animation-style images. Their work shows that domain-specific feature learning, attuned to artistic styles, can greatly improve classification accuracy. For our project, these findings suggest that adapting our model to handle stylized features such as color palettes and texture styles, which are critical for Pokémon type identification, where visual elements like flames, water waves, or electrical sparks often signal underlying types.

3.7 Conclusion of Findings

Through this review, we have identified a range of techniques and insights that can directly inform our approach to Pokémon type classification. CNNs, hybrid CNN-RNN models, transformer architectures, transfer learning strategies, and domain-specific adaptations all offer valuable tools for tackling the challenges of our project. By integrating these findings, we aim to design a model that is flexible, robust, and accurate, one that embraces the creative complexity of Pokémon designs while applying state-of-the-art machine learning techniques.

4 Methodology

The goal of classifying Pokémon into their primary types is a supervised image classification problem, which means that every image is associated with a label corresponding to its primary type. Models were trained using cross-entropy loss, the Adam optimizer, and evaluated via accuracy, F1 score, top-5 accuracy, and confusion matrices.

4.1 Dataset

Our dataset was pulled from Kaggle and initially had 26,000 128 x 128 images of 1,000 Pokémon. To prevent data leakage and ensure the model does not simply memorize individual Pokémon, we performed the data split by randomly assigning entire Pokémon species to the training, validation, and test sets. One major problem that was encountered was with data balance since this dataset was very imbalanced. One especially egregious example of this was with the flying types which only had around 100 samples across 5 different Pokémon. This was simply too little data to train on and oversampling was not an option since we were splitting by Pokémon species, not just individual images. As a result, we decided to remove

this type from all processes entirely. As mentioned before, a train validation test split was used at a ratio of 0.8, 0.1, and 0.1 respectively.

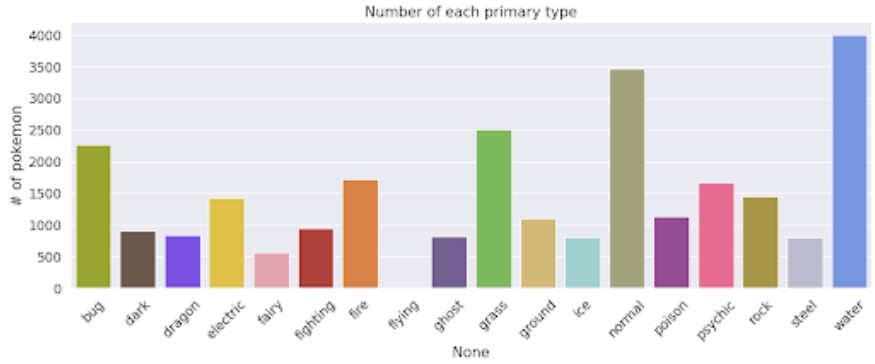

Figure 1: Shows the number of Pokémon associated with each primary elemental type in the dataset

4.2 Representation

Each image is represented as a three dimensional tensor with shape of H x W x C where H is the height of the image, W is the width of the image, and C is the number of channels. In our case, images have a height and width of 128 with 3 channels to represent RGB. This results in a default shape of 128 x 128 x 3. All images were also normalized from a [0, 255] scale to a [0, 1] scale.

4.3 Models

4.3.1 Multi Layer Perceptron (MLP)

An MLP was used as a baseline approach to this problem. The images are flattened into a one dimensional vectors before being passed into the model. Due to memory constraints, the images have to be resized to 64 x 64 prior to being passed into the model. Then they are passed through multiple linear layers with ReLus as the nonlinearity functions between layers. Dropout layers were also added to attempt to avoid overfitting.

MLP Model 1 Architecture

- **Input:** `12,288` (flattened image)
- **Linear layer:** `12,288 → 2,048`, activation: ReLU
- **Dropout layer**
- **Linear layer:** `2,048 → 512`, activation: ReLU

- **Dropout layer**
- **Output layer:** `512 → 17`, activation: Softmax
- **Learning rate:** `0.001`, Batch size: `100`, Epochs: `10`

MLP Model 2 Architecture

- **Input:** `12,288` (flattened image)
- **Linear layer:** `12,288 → 1,024`, activation: ReLU
- **Dropout layer**
- **Linear layer:** `1,024 → 512`, activation: ReLU
- **Dropout layer**
- **Linear layer:** `512 → 128`, activation: ReLU
- **Output layer:** `128 → 17`, activation: Softmax
- **Learning rate:** `0.001`, Batch size: `100`, Epochs: `10`

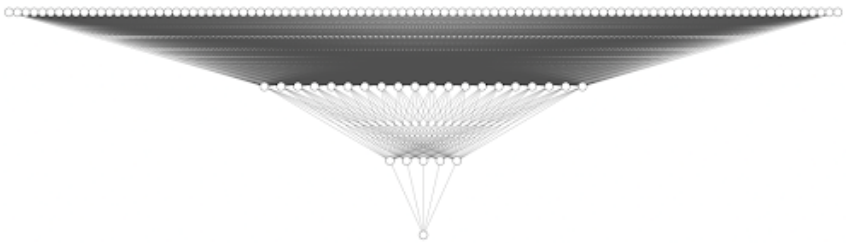

Figure 2: Illustrates the architecture for the MLP 1 and MLP 2 models utilized in this project.

4.3.2 Convolutional Neural Network (CNN)

A CNN was implemented to attempt to exploit the innate spatial relationships within images and learn from raw pixel data without requiring manual feature engineering. The architecture consisted of convolutional layers with ReLU activations and max-pooling, followed by linear layers.

CNN Model Architecture

- **Input tensor:** `128 × 128 × 3`
- **Conv2D Layer 1:** Kernel size = `3 × 3`, Depth = `10` ⇒ `126 × 126 × 10`
- **Max Pooling Layer 1:** Pool size = `2 × 2` ⇒ `63 × 63 × 10`
- **Conv2D Layer 2:** Kernel size = `3 × 3`, Depth = `16` ⇒ `61 × 61 × 16`
- **Max Pooling Layer 2:** Pool size = `2 × 2` ⇒ `30 × 30 × 16`
- **Conv2D Layer 3:** Kernel size = `3 × 3`, Depth = `16` ⇒ `28 × 28 × 16`
- **Flatten:** `28 × 28 × 16 = 12,544`
- **Linear layer 1:** `12,544 → 4,000`, activation: ReLU
- **Dropout layer**
- **Linear layer 2:** `4,000 → 512`, activation: ReLU
- **Output layer:** `512 → 17`, activation: Softmax
- **Learning rate:** `0.001`, Batch size: `100`, Epochs: `10`

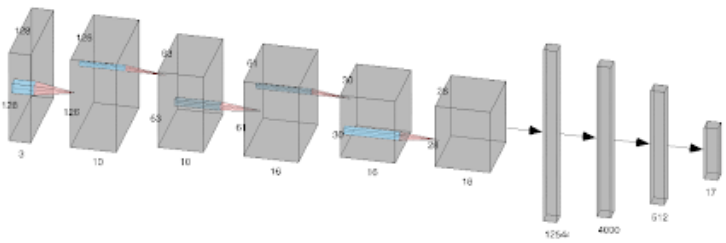

Figure 3: Illustrates the architecture for the CNN model utilized in this project.

5 Experimental Design and Results

Throughout the course of this project, we trained and evaluated four models, two MLPs and two CNNs, on our stylized Pokémon dataset. Unfortunately, due to hardware limitations and long training times, we were unable to fully train CNN 2. As a result, its performance was not included in our final comparison. These constraints reflect the reality of working with limited computational resources on personal machines, especially within the time frame of a semester-long class project. Among the remaining models, CNN 1 achieved the highest training accuracy, reaching 84.2%. This confirmed its strength in extracting detailed visual patterns from images. However, the model's test accuracy dropped sharply to 18.1%, revealing its struggle to generalize beyond the training data. This kind of overfitting is a common issue when models are highly complex but trained on limited or imbalanced datasets.

Metric	**MLP 1**	**MLP 2**	**CNN 1**
Train Accuracy (%)	53.0	50.8	84.2
Validation Accuracy (%)	24.1	19.8	21.5
Test Accuracy (%)	22.3	20.8	18.1
F1 Score	0.1149	0.1143	0.1251
Top 5 Accuracy	0.4627	0.4169	0.4698

Table 1: Comparison of MLP and CNN models across multiple metrics

In contrast, MLP 1 delivered the best performance on both the validation and test sets, with 24.1% and 22.3% accuracy. Despite being less sophisticated than CNNs, its simpler architecture helped avoid overfitting and allowed for more consistent performance across different data splits. MLP 2 followed closely with slightly lower scores. Interestingly, CNN 1 achieved the highest top-5 accuracy at 46.98%, which suggests that even when it failed to predict the correct label as the top choice, it often ranked it among its top guesses, a sign that it was learning useful, though not always decisive, features. F1 scores for all models were relatively low, ranging from 0.1143 (MLP 2) to 0.1251 (CNN 1). This metric is particularly sensitive to how well models perform across all classes, including the underrepresented ones. In our dataset, types like Water and Normal appeared frequently, while types like Fairy and Flying were rare or removed entirely due to insufficient samples. This imbalance made it difficult for the models to accurately predict those less common types and likely contributed to the lower F1 scores and misclassifications.

Overall, these results reflect the core challenges of working with stylized data, limited training resources, and class imbalance. While MLPs demonstrated surprising resilience, CNNs showed potential if given more data and regularization. The combination of top-1 and top-5 scores also hints at how different evaluation strategies can provide a more complete picture of model behavior.

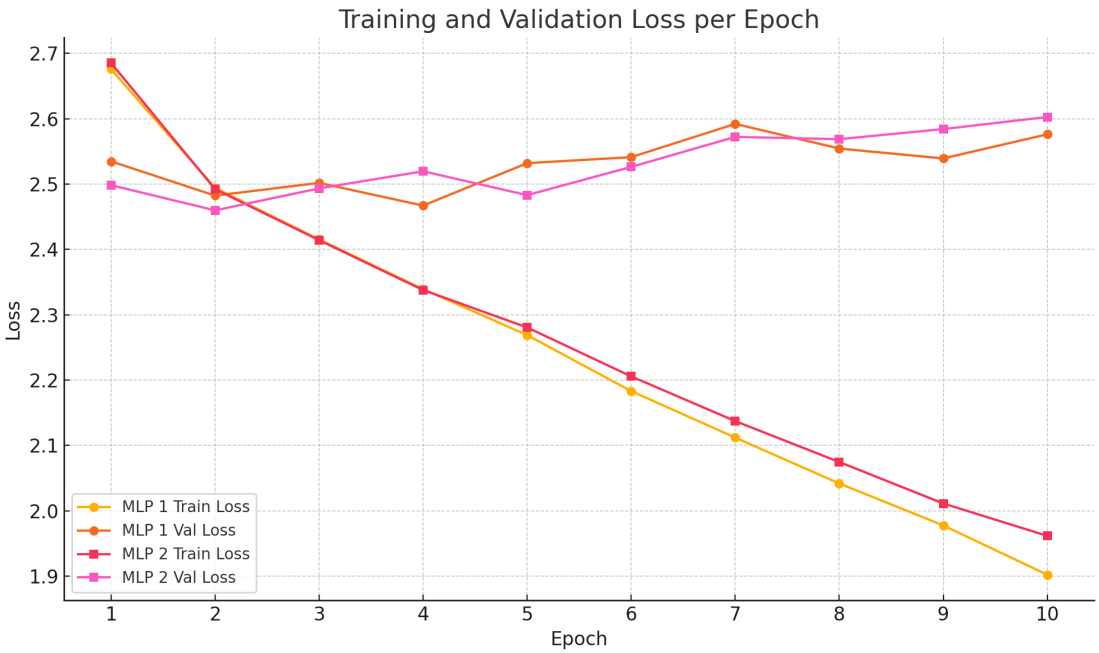

Figure 4: MLP Training Validation Loss

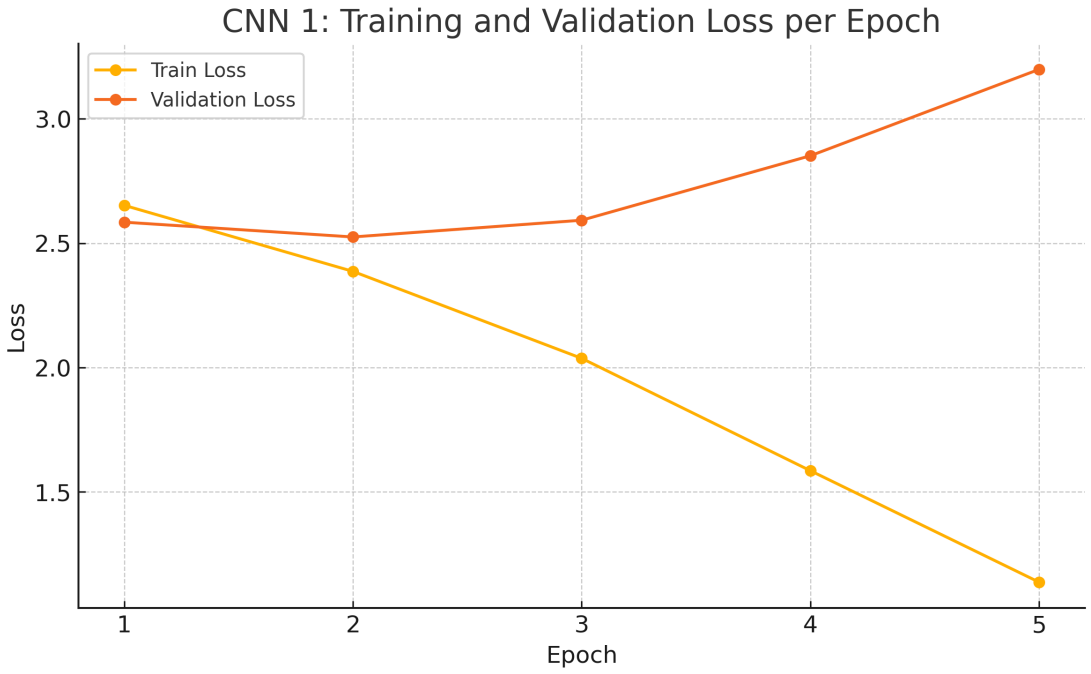

Figure 5: CNN 1 Training Validation Loss

6 Analysis

Our results reveal important trade-offs between model complexity and generalization. While CNN 1 demonstrated strong training performance, it suffered from overfitting, likely due to limited training data and class imbalance. MLP 1, while less complex, generalized better to the validation and test sets. The consistently low F1 scores suggest that all models struggled with minority class detection, which is typical in imbalanced classification problems. The high top-5 accuracy scores indicate that while models may not have predicted the exact label correctly, they often ranked it among the top few options. This is a positive sign for future improvements.

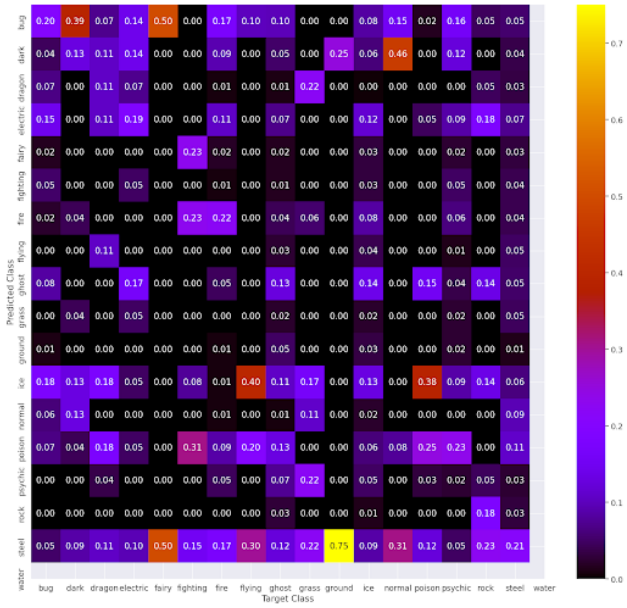

Figure 6: Provides confusion matrices for MLP models, demonstrating typical misclassifications due to class imbalance and dataset variability

The stylized nature of the dataset likely added to the difficulty, as visual features may be less consistent or realistic than in standard datasets. Regularization techniques, data augmentation, and better handling of class imbalance (e.g., weighted loss functions or oversampling) are potential next steps. Moreover, incorporating pretrained vision models or experimenting with transformer-based architectures could yield better results.

7 Conclusion and Future Work

This project compared MLPs and CNNs for the task of Pokémon type classification using stylized images. While CNNs showed stronger feature extraction, they were prone to overfitting. MLPs, though simpler, performed better on unseen data in some cases. Performance overall was limited by dataset imbalance and the nature of the image inputs. Alongside

training our own models, we also looked into models that others have already trained. One that stood out was the kuhs/pokemon-vit Vision Transformer, hosted on Hugging Face. It was already fine-tuned on a Pokémon dataset and reached a validation accuracy of 68.4% after just five training rounds. This shows a lot of promise, especially since it's trained on similar data and doesn't need to start from scratch like our models did. Using a pretrained model like this has a lot of appeal. Instead of training a full model ourselves, we could use it as a base and build a simpler layer on top of it, or even use it as-is to predict types from new images. This kind of approach is especially useful when working with limited time or resources. As this was a class project, we were under tight time constraints and limited by the computing power available on our personal machines, which made it difficult to train deeper or more complex models from scratch. It's also exciting to see that transformer-based models, which are relatively new to image classification, can do so well on something as specific and stylized as Pokémon. Moving forward, we plan to test this pretrained model more directly in our own pipeline. We're also interested in trying lightweight tuning methods or combining different models, like using both transformers and CNNs together. In addition, we want to improve how we handle type imbalance in our dataset and explore more creative data augmentation strategies. Our long-term goal is to build a model that can understand and classify stylized content more reliably, and maybe even adapt across different artistic styles or visual themes.

8 Contributions

Vy completed sections 1, 3, 4.2 - Liem completed sections 2, 3, 4.1 - Quenton completed 5, 6, 7

References

[1] Paolo Didier Alfano et al. "Top-tuning: A study on transfer learning for an efficient alternative to fine tuning for image classification with fast kernel methods". In: *Image and Vision Computing* 142 (2024), p. 104894. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2023.104894. URL: https://www.sciencedirect.com/science/article/pii/S0262885623002688.

[2] Sai Yeshwanth Chaganti et al. "Image Classification using SVM and CNN". In: *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*. 2020, pp. 1–5. DOI: 10.1109/ICCSEA49143.2020.9132851.

[3] Patel Dhruv and Subham Naskar. "Image Classification Using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN): A Review". In: *Machine Learning and Information Processing*. Ed. by Debabala Swain, Prasant Kumar Pattnaik, and Pradeep K. Gupta. Singapore: Springer Singapore, 2020, pp. 367–381. ISBN: 978-981-15-1884-3.

[4] Norliza Endut et al. "A Systematic Literature Review on Multi-Label Classification Based on Machine Learning Algorithms". In: *TEM Journal* 11.2 (2022), pp. 658–666. URL: https://www.temjournal.com/content/112/TEMJournalMay2022_658_666.pdf.

[5] Shanqing Gu, Manisha Pednekar, and Robert Slater. "Improve image classification using data augmentation and neural networks". In: *SMU Data Science Review* 2.2 (2019), p. 1.

[6] Nandhini Santhanam Hee E. Kim Alejandro Cosa-Linan. "Transfer learning for medical image classification: a literature review". In: *BMC Medical Imaging* 22.1 (2022), pp. 1–13. DOI: 10.1186/s12880-022-00793-7. URL: https://doi.org/10.1186/s12880-022-00793-7.

[7] Navdeep Kaur, Dilbag Singh, and Ashish Arora. "Synthetic Image Generation Using Deep Learning: A Systematic Review". In: *Computational Intelligence* (2023). DOI: 10.1111/coin.70002. URL: https://onlinelibrary.wiley.com/doi/10.1111/coin.70002%7D.

[8] Alina Kuznetsova et al. "The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale". eng. In: *International journal of computer vision* 128.7 (2020), pp. 1956–1981. ISSN: 0920-5691.

[9] Jack Lanchantin et al. "General Multi-Label Image Classification With Transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 16478–16488.

[10] Jiali Lyu et al. "Color Matching Generation Algorithm for Animation Characters Based on Convolutional Neural Network". eng. In: *Computational intelligence and neuroscience* 2022 (2022), pp. 1–13. ISSN: 1687-5265.

[11] Rahul Mishra and Arnav Bhavsar. "Analyzing Image Classification via EEG". In: *Computer Vision, Pattern Recognition, Image Processing, and Graphics*. Ed. by R. Venkatesh Babu, Mahadeva Prasanna, and Vinay P. Namboodiri. Singapore: Springer Singapore, 2020, pp. 537–547. ISBN: 978-981-15-8697-2.

[12] Jo Plested and Tom Gedeon. *Deep transfer learning for image classification: a survey*. 2022. arXiv: 2205.09904 [cs.CV]. URL: https://arxiv.org/abs/2205.09904.

[13] Isaac Triguero and Celine Vens. "Labelling strategies for hierarchical multi-label classification techniques". In: *Pattern Recognition* 56 (2016), pp. 170–183. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2016.02.017. URL: https://www.sciencedirect.com/science/article/pii/S0031320316000881.

[14] Dong Wang, Jian Lian, and Wanzhen Jiao. "Multi-label classification of retinal disease via a novel vision transformer model". In: *Frontiers in Neuroscience* Volume 17 - 2023 (2024). ISSN: 1662-453X. DOI: 10.3389/fnins.2023.1290803. URL: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2023.1290803.

[15] Jiang Wang et al. " CNN-RNN: A Unified Framework for Multi-label Image Classification ". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2016, pp. 2285–2294. DOI: 10.1109/CVPR.2016.251. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.251.

[16] Yaoli Wang et al. "Vision Transformers for Image Classification: A Comparative Survey". In: *Technologies* 13.1 (2025). ISSN: 2227-7080. DOI: 10.3390/technologies13010032. URL: https://www.mdpi.com/2227-7080/13/1/32.