Brazilian Jiu-Jitsu Image Recognition using Support Vector Machines, Convolutional Neural Network, and Pre-trained Models with Transfer-learning

Mingi Lee, Sergio Martelo, Masaki Nawa, and Ryu Sonoda

Abstract

We present four machine learning models for classifying Brazilian Jiu-Jitsu (BJJ) positions from images: a Support Vector Machine (SVM), a Convolutional Neural Network (CNN), and transfer learning using a pre-trained model (RegNet). We trained the models on a dataset of 120,279 labeled images of two BJJ athletes in 10 distinct positions. Our CNN model achieved an accuracy of 97%, comparable to our fine-tuned and feature-extracted RegNet models' 98.7% and 99.5%, but required less training time. The SVM performed worse at 96.4% accuracy. The models struggled to distinguish between the takedown position and standing position, frequently misclassifying the former as the latter. Our models are limited by overfitting to the dataset and may not generalize to new data, but show promise for automated BJJ position recognition. Larger, diverse datasets and more computation power could enable reliable and fair scoring systems to benefit the sport.

Code and data set can be accessed through the link below:

https://github.com/sonodary/BJJ_machine_learning

Introduction

In the past decade, Machine Learning as a field has made incredible leaps and bounds in terms of achievements and accessibility thanks to advances in computer hardware and data collection (Brynjolfsson 2017). Image recognition is one of many fields that have sprung from this seemingly unstoppable rise and have the potential to change the world as we know it. From self-driving cars to face recognition at a mass scale, image recognition with machine learning has a myriad of potential applications (Yasar 2023). The application we focus on, although simpler, is also interesting: The recognition of Brazilian Jiu-Jitsu positions in images of two people practicing the sport.

Brazilian Jiu-Jitsu, or BJJ for short, is one of the fastest growing sports in the United States. Going from relative obscurity in the 1990's to being the second most practiced combat sport in the U.S. behind only boxing (Millar 2021). As a submission grappling sport, BJJ is focused on transitioning from positions that vary in dominance to other, hopefully more dominant, positions against resisting opponents with the goal of applying submission holds and making the opponent surrender. The task of identifying BJJ positions is one of importance to the sport because, in the absence of a definitive surrender by one of the combatants, the outcome of a match is decided by which combatant was able to get to more dominant positions (Medeiros 2019). As of now, positions and points are decided by a single referee, but we believe that an automated system that uses image recognition would be a more stream-lined and less error-prone process that would lead to increased fairness in the sport.

To perform this position recognition, we built four different machine learning models: a Support Vector Machine, a Convolutional Neural Network, and performed transfer learning using two different approaches and a widely recognized pre-trained model (Xu et al. 2021). We also use a dataset of images of two BJJ athletes sparring in 10 distinct positions that have been labeled by the authors of (Hudovernik et al. 2022). Our methods differ from those of (Hudovernik et al. 2022) in that we use a more generalized approach that does not rely on key-point and person tracking. We also differ in that we focus on intaking single images rather than video. This, however, is due to our computational and time constraints.

As stated above, we present four different models for classifying BJJ positions from images of two athletes fighting. All four of our models were accurate at this task, but our Neural Networks outperformed our baseline model by a wide margin. Also, our simpler model's performance was comparable to that of our fine-tuned and feature-extracted pre-trained models while needing a fraction of the computational power to train. We further discuss this Accuracy-Complexity trade-off in the discussion section below.

We begin by providing an overview of our methods and all three models that we used. We then provide our results and a discussion on our takeaways and limitations.

Methods

The data set used in this paper was collected from the Visual Cognitive Systems Laboratory at the University of Ljubljana Faculty of Computer and Information Science. The dataset consists of 120,279 labeled images of two Brazilian Jiu-Jitsu athletes sparring in different combat positions, with a total of 10 positions and 18 classes. Further details about each variable are presented in Table 1. Since the labels in our dataset were strings and the model's output requires numeric variables, we mapped each label to a number between 0 and 17. Due to the large size of the training data, it was not feasible to store all the images on GitHub. Instead, we uploaded 12 batches of compressed datasets, each containing 10,000 images, to Google Drive. To download and process these images faster, we developed our own function that utilized multiprocessing. We downloaded the zipped files and uncompressed them into 64x64 images. After obtaining the images, we divided the dataset into training and testing sets, with sizes of 75% and 25% of the original dataset, respectively. The testing dataset was initially trained on a Support Vector Machine (SVM) to serve as our benchmark model. We chose SVM as a benchmark because it is a relatively simple model for image classification and requires less training time.

As our main model, we developed a customized Convolutional Neural Network (CNN). We started with a small base model consisting of two convolutional layers, each followed by a max-pooling layer. The final two layers were fully connected layers, mapping to 18 classes. ReLU was used as the activation function. To optimize the hyperparameters such as kernel size and max-pooling size, we trained the model using three batches. Based on the results, we decided to use a kernel size of 3x3 with a stride of 1 and no padding, and a max-pooling size of 2x2. We then prepared two additional models: medium and large architectures. The medium model had three convolutional layers, while the large model had four convolutional layers, both followed by max-pooling. The remaining layers were identical to the small model. Again, we trained each model using three batches to determine the best-performing model. As a result, we selected the large architecture. The final model consists of four convolutional layers with a kernel size of 3x3 (the second convolutional layer has 2x2 kernel size), a stride of 1, and no padding. The first three layers are followed by 2x2 max-pooling layers. The last two layers are fully connected layers, with dimensions of 1024 to 200 and 200 to 18. ReLU is used as the activation function (see Figure 1).

To assess the performance of our customized model compared to an existing model, we trained our dataset on a pre-trained RegNet using both feature extraction and fine-tuning approaches. RegNet is a neural network architecture proposed by the Facebook AI team to achieve a more flexible architecture. It was trained on ImageNet, which includes 1000 classes of images such as humans, automobiles, and animals. RegNet_Y_800 achieved higher accuracy than EfficientNet with a shorter training time (Radosavovic et al. 2020, 10). Among multiple versions of RegNet, we chose RegNet_X_400 as it is a relatively small architecture with 4 million parameters. For both fine-tuning and feature extraction, since the output of RegNet is the likelihood of the input image being classified into 1000 classes, we replaced the final layer of RegNet so that the output represents the likelihood of 18 classes.

Table 1

Descriptions and Names of Combat Positions

Positions	Description
Standing	Two athletes standing but in no combat position; labeled as standing

Takedown	2 classes designating which athlete is initiating the takedown; if athlete 1 is initiating the takedown, it is labeled as takedown1
Open guard	2 classes designating which athlete is currently in the guard position; if athlete 1 is in the guard position, it is labeled as open_guard1
Half guard	2 classes designating which athlete is currently in the half-guard position; if athlete 1 is in the half-guard position, it is labeled as half_guard1
Closed guard	2 classes designating which athlete is currently in the closed-guard position; if athlete 1 is in the closed-guard position, it is labeled as closed_guard1
50-50 guard	Both athletes in the guard position; labeled as 5050_guard
Side control	2 classes designating which athlete is in top side control position; if athlete 1 is in the top side position, it is labeled as side_control1
Mount	2 classes designating which athlete is in top mount position; if athlete 1 is in the top mount position, it is labeled as mount1
Back	2 classes designating which athlete is controlling the opponent's back; if athlete 1 is controlling the opponent's back, it is labeled as back1
Turtle	2 classes designating which athlete is in turtle position; if athlete 1 is in the turtle position, it is labeled as turtle1

Results

In this section, we present the results obtained from applying each of the models to the data set. We provide cost plots by training epoch to verify that the model has converged, various evaluation metrics, and confusion matrices to determine how our models perform at classifying each position.

Support Vector Machine

As a benchmark model, we trained a basic Support Vector Machine with the dataset to predict the positions of the athletes. We applied standard scaling and used a stochastic gradient descent classifier to train the SVM model on the training set.

Figure 3a shows the evaluation metrics for the SVM model. All of the metrics are lower than those of the other models, and therefore we conclude that the Support Vector Machine is a worse classifier for our data than the other models we created. However, the SVM model took approximately an hour to train, which is by far the least amount of time amongst our models. We do not think this trade-off is valuable, however, because we do believe that this model would perform far worse on new data than the other models.

Figure 3b shows the confusion matrix of the SVM model, from which we can observe that it is frequently misclassifying takedown positions 1 and 2. Furthermore, this model tends to misclassify other positions like position 8 (closed_guard2), position 16, and position 17 (side control), which our other models were highly accurate at classifying.

The reason the SVM model underperforms compared to the other models could be a topic for future discussion. Potential reasons include the nonlinear nature of some positions which the linear SVM may not be able to model accurately, as well as the imbalance in class sizes which may bias the SVM training. The model also lacks the flexibility that makes Convolutional Neural Networks so successful at classifying images. More specifically, the fact that the features SVMs rely on for their classification are location dependent within the frame makes them inferior to CNNs at identifying dynamic positions or positions in images with strange formatting.

Convolutional Neural Network

We developed a custom convolutional neural network to predict the positions. We split the data set into training and test data sets with a test size of 0.25. The hyperparameters utilized for this model were an epoch size of 300, a learning rate of 0.001, and a batch size of 32. We employed the Stochastic Gradient Descent optimizer when training the model. The training process took approximately 8 hours.

Figure 4a shows the cost for each training epoch, which steadily decreased and eventually converged. This indicates the hyperparameters chosen were effective. Figure 4b displays various evaluation metrics on the testing data. We included the classification accuracy score, F1 score, and balanced accuracy score due to the imbalanced class size as seen in Figure 2. All evaluation metrics had high scores near 1, indicating the model performed well on the testing data.

To determine if some positions were harder to classify, we generated a confusion matrix with row-wise normalization. Figure 4c shows that the most commonly misclassified images were positions 1 and 2 (takedown). The model misclassified approximately 15% of positions 1 and 2 as position 0 (standing). We believe this occurred because both players were standing in some takedown images, causing the model to misclassify the positions. Modifying the CNN architecture may address this issue and serve as future research.

Transfer Learning

First, we utilized fine-tuned transfer learning to predict the positions using RegNet. We employed the same parameters as the CNN for the training and testing data split with a test size of 0.25. For the hyperparameters, we used an epoch size of 150, a learning rate of 0.0001, and a batch size of 32 after testing different values and verifying the cost plot. We used the Adam optimizer to train the data set. It took approximately 40 hours to complete the training.

Figure 5a shows that the cost of the model began converging around the 20th epoch, validating our learning rate and epoch size. Figure 5b shows different evaluation metrics for this model. We see that the fine-tuned approach of transfer learning using RegNet had better scores than our CNN model with an accuracy score of 0.987.

We also used RegNet with a feature extraction approach to compare performance with other models. We used the same number of epochs and batch size but changed the learning rate to 0.001 as 0.0001 was needlessly small for this approach. The feature extraction model took about 28 hours, which is shorter than the fine-tuning approach but still longer than the CNN model.

Figure 6a shows that the feature extraction approach with a learning rate of 0.001 also began converging around the 20th epoch. The feature extraction approach generated the best scores among all models with an accuracy score of 0.995 as we can see from Figure 6b. We concluded that feature extraction is more suitable for this data set as it took less time but produced a better result.

Discussion and Limitations

As we can see above, our customized Convolutional Neural Network (CNN) outperformed our benchmark model, the Support Vector Machine, but failed to outperform pre-trained models adjusted to this task by using various approaches to transfer learning. This can be seen as an obvious conclusion since RegNet, the pre-trained model we used as our base for transfer learning, has over 5 million parameters trained on ImageNet, a dataset of 1.2 million images. It seems trivial then to conclude that this state-of-the-art model would outperform our much smaller CNN trained on a tenth of the images. However, as shown above, our custom CNN only performed marginally worse than our transfer learning models and took much less time to train. We believe this trade-off between training time and accuracy is a worthwhile one, specially in the absence of significant computational power.

Most of our models struggled with classifying the takedown position and often misclassified it as the standing position. We believe this error can be attributed to the similarity between the early stages of a takedown and the standing position. Takedowns are normally initiated from the standing position and, as such, delineating the start of a takedown and the end of the standing position can be hard and is oftentimes subjective.

Many of the limitations of our models come from the nature of our dataset. Although we had a relatively large number of images, all of these images were of the same two athletes in roughly the same location and position within the frame. This means that while our models were highly accurate at classifying these images, they might be over-tuned to the particularities of these two athletes and their setting. Thus, we would expect to see a hefty decline in performance on outside data. Thankfully our models had such high accuracy that we believe that, even with a hefty decline in performance, they would still prove to be accurate at classifying BJJ positions.

Other limitations that arise from the dataset, are that both of the athletes are not wearing Gis. Gis are the traditional uniforms of BJJ, they are shared with many other martial arts and are widely used while training the sport. Not wearing a Gi is also pretty widespread, however, and, thus, the athletes decided to not wear it. We believe that the Gi provides an extra challenge for image recognition models because it obscures many of the athletes features and could make it difficult for models to differentiate between combatants, specially when the Gis they are wearing are the same color.

This then means that our models will be unable to generalize their current features to pictures of people sparring while wearing Gis. An extension of our work could be to train the models on datasets containing both types of images and maybe even taking images of people wearing Gis going against people not wearing Gis, although this is not a common situation in the sport.

Additionally, we also believe our models would have a hard time with images that contain more than two people, such as images of two people sparring while others walk in the background. This weakness also arises from our dataset and could be fixed by adding new images to our current set.

Finally, one aim of this paper that we did not have time to carry out is the validation of our models with our own images of various people sparring here at the college. We believe this would serve as a good test for our models, however, due to college policy, the club had to stop meeting before we were able to take the images.

This paper shows the potential of applying advanced neural networks and transfer learning to identify Brazilian Jiu-Jitsu positions. Future research could make use of larger datasets and more computation power to create systems/models that could benefit the sport by accurately recognizing positions despite external factors. This would result in more reliable, error-free scoring that promotes fairness in the sport. Furthermore, we believe this would also make running a competition more accessible to organizations in smaller communities with less of an interest in BJJ, thereby making the sport more accessible as a whole.

References

Brynjolfsson and McAfee. “What’s Driving the Machine Learning Explosion?” 2017. Harvard Business Review. July 18, 2017. https://hbr.org/2017/07/whats-driving-the-machine-learning-explosion.

Hudovernik, Valter, and Danijel Skocaj. “Video-Based Detection of Combat Positions and Automatic Scoring in Jiu-Jitsu.” Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports. ACM, October 10, 2022. https://doi.org/10.1145/3552437.3555707.

Medeiros. “9 Golden Rules of Jiu-Jitsu.” 2019. Gracie Barra. December 30, 2019. https://graciebarra.com/gb-news/9-golden-rules-of-jiu-jitsu/.

Millar. “The Rise and Rise of Brazilian Jiu-Jitsu.” 2021. Men’s Health. July 11, 2021. https://www.menshealth.com/uk/mhsquad/big-reads-membership/a26808956/brazilian-jiu-jitsu-guide/.

Radosavovic, Ilija, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. “Designing Network Design Spaces.” arXiv.org, March 30, 2020. https://arxiv.org/abs/2003.13678.

Xu, Jing, Yu Pan, Xinglin Pan, Steven Hoi, Zhang Yi, and Zenglin Xu. ‘RegNet: Self-Regulated Network for Image Classification’. ArXiv [Eess.IV], 3 January 2021. arXiv. http://arxiv.org/abs/2101.00590.

Yasar. “What Is Image Recognition? | Definition from TechTarget.” 2023. Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/image-recognition#:~:text=Image%20recognition%20is%20used%20to.

Appendix A

Fig. 1 Architecture of customized CNN (Fully connected layers are omitted)

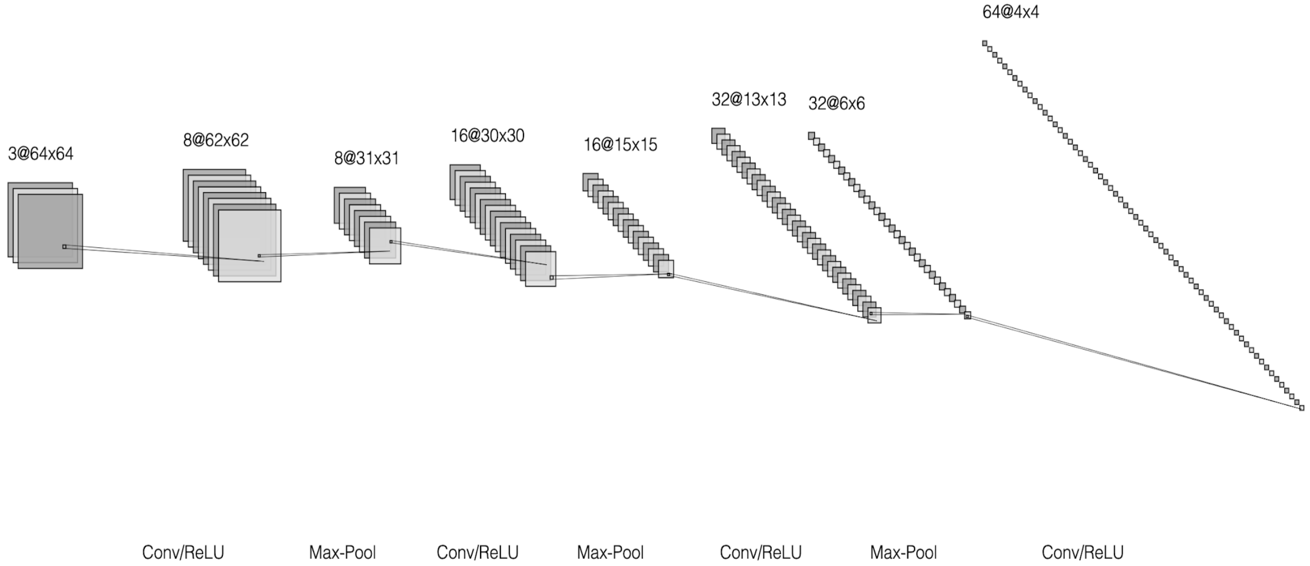
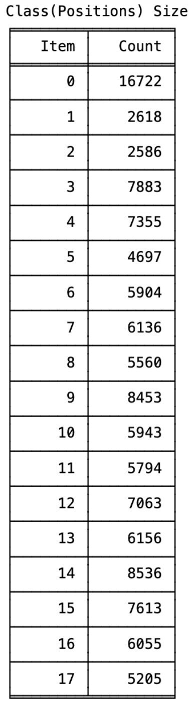


Fig. 2 The Size of Each Class (Positions)

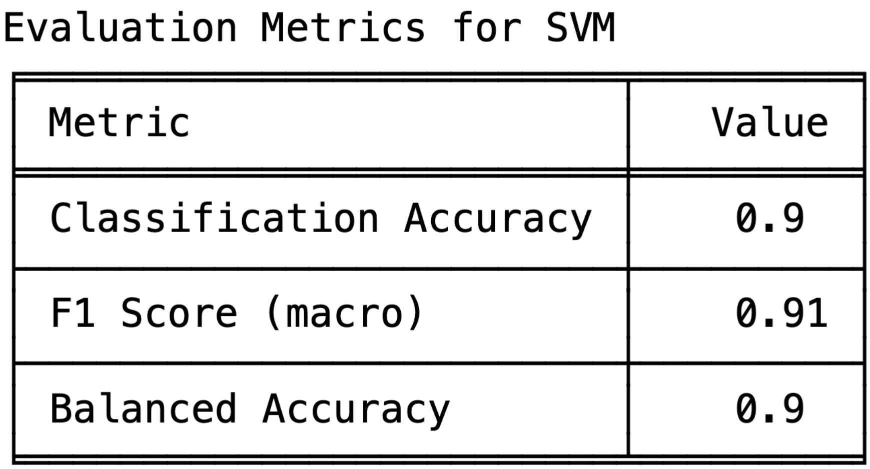


Class(Positions) Size

Item	Count
0	16722
1	2618
2	2586
3	7883
4	7355
5	4697
6	5904
7	6136
8	5560
9	8453
10	5943
11	5794
12	7063
13	6156
14	8536
15	7613
16	6055
17	5205

Fig. 3

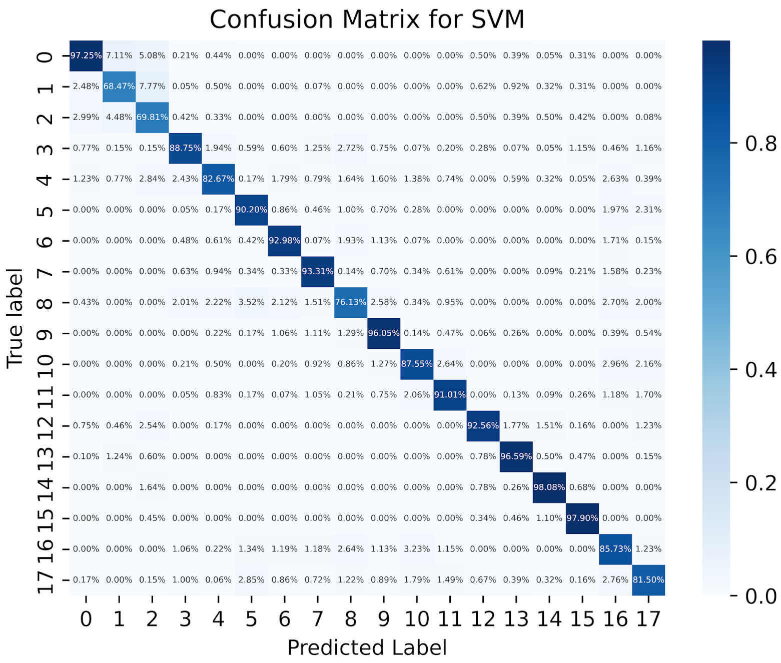
A) Evaluation Metrics for SVM



Evaluation Metrics for SVM

Metric	Value
Classification Accuracy	0.9
F1 Score (macro)	0.91
Balanced Accuracy	0.9

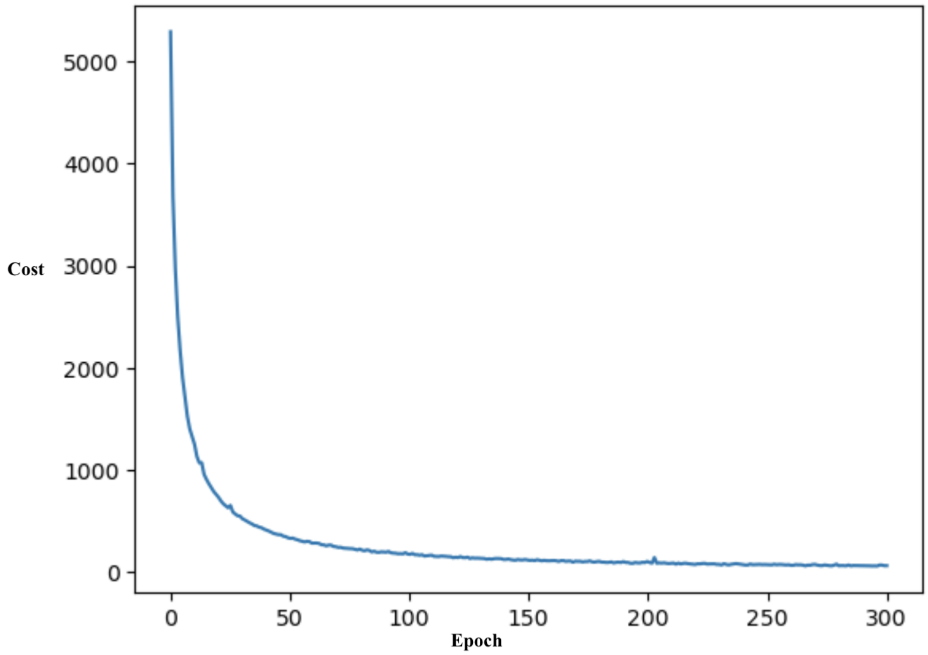
B) Confusion Matrix for SVM



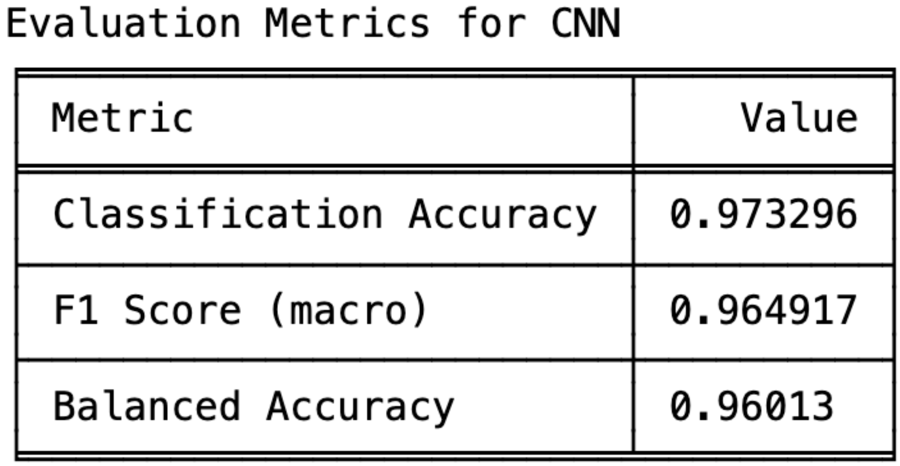
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	97.25%	7.11%	5.08%	0.21%	0.44%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.50%	0.39%	0.05%	0.31%	0.00%	0.00%
1	2.48%	68.47%	7.77%	0.05%	0.50%	0.00%	0.00%	0.07%	0.00%	0.00%	0.00%	0.00%	0.62%	0.92%	0.32%	0.31%	0.00%	0.00%
2	2.99%	4.48%	69.81%	0.42%	0.33%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.50%	0.39%	0.50%	0.42%	0.00%	0.08%
3	0.77%	0.15%	0.15%	88.75%	1.94%	0.59%	0.60%	1.25%	2.72%	0.75%	0.07%	0.20%	0.28%	0.07%	0.05%	1.15%	0.46%	1.16%
4	1.23%	0.77%	2.84%	2.43%	82.67%	0.17%	1.79%	0.79%	1.64%	1.60%	1.38%	0.74%	0.00%	0.59%	0.32%	0.05%	2.63%	0.39%
5	0.00%	0.00%	0.00%	0.05%	0.17%	90.20%	0.86%	0.46%	1.00%	0.70%	0.28%	0.00%	0.00%	0.00%	0.00%	0.00%	1.97%	2.31%
6	0.00%	0.00%	0.00%	0.48%	0.61%	0.42%	92.98%	0.07%	1.93%	1.13%	0.07%	0.00%	0.00%	0.00%	0.00%	0.00%	1.71%	0.15%
7	0.00%	0.00%	0.00%	0.63%	0.94%	0.34%	0.33%	93.31%	0.14%	0.70%	0.34%	0.61%	0.00%	0.00%	0.09%	0.21%	1.58%	0.23%
8	0.43%	0.00%	0.00%	2.01%	2.22%	3.52%	2.12%	1.51%	76.13%	2.58%	0.34%	0.95%	0.00%	0.00%	0.00%	0.00%	2.70%	2.00%
9	0.00%	0.00%	0.00%	0.00%	0.22%	0.17%	1.06%	1.11%	1.29%	96.05%	0.14%	0.47%	0.06%	0.26%	0.00%	0.00%	0.39%	0.54%
10	0.00%	0.00%	0.00%	0.21%	0.50%	0.00%	0.20%	0.92%	0.86%	1.27%	87.55%	2.64%	0.00%	0.00%	0.00%	0.00%	2.96%	2.16%
11	0.00%	0.00%	0.00%	0.05%	0.83%	0.17%	0.07%	1.05%	0.21%	0.75%	2.06%	91.01%	0.00%	0.13%	0.09%	0.26%	1.18%	1.70%
12	0.75%	0.46%	2.54%	0.00%	0.17%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	92.56%	1.77%	1.51%	0.16%	0.00%	1.23%
13	0.10%	1.24%	0.60%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.78%	96.59%	0.50%	0.47%	0.00%	0.15%
14	0.00%	0.00%	1.64%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.78%	0.26%	98.08%	0.68%	0.00%	0.00%
15	0.00%	0.00%	0.45%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.34%	0.46%	1.10%	97.90%	0.00%	0.00%
16	0.00%	0.00%	0.00%	1.06%	0.22%	1.34%	1.19%	1.18%	2.64%	1.13%	3.23%	1.15%	0.00%	0.00%	0.00%	0.00%	85.73%	1.23%
17	0.17%	0.00%	0.15%	1.00%	0.06%	2.85%	0.86%	0.72%	1.22%	0.89%	1.79%	1.49%	0.67%	0.39%	0.32%	0.16%	2.76%	81.50%

Fig. 4

A) Cost Plot for CNN



B) Evaluation Metrics for CNN



Evaluation Metrics for CNN

Metric	Value
Classification Accuracy	0.973296
F1 Score (macro)	0.964917
Balanced Accuracy	0.96013

C) Confusion Matrix for CNN

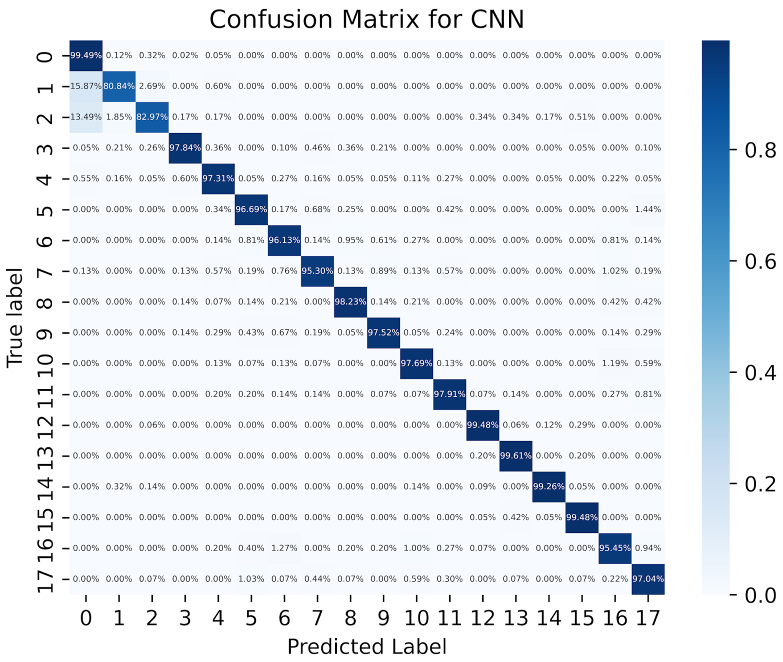
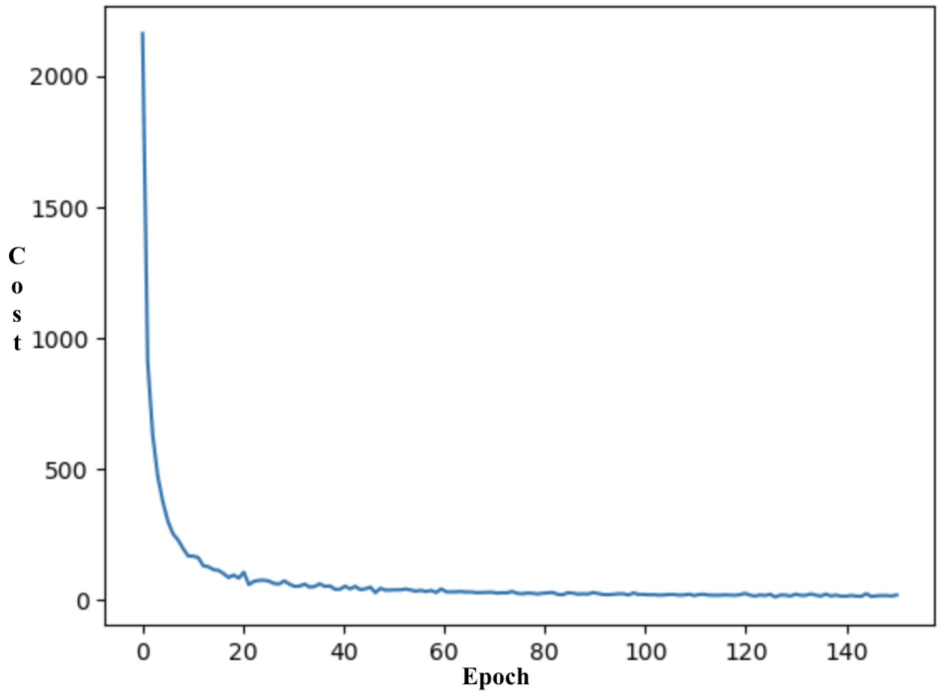
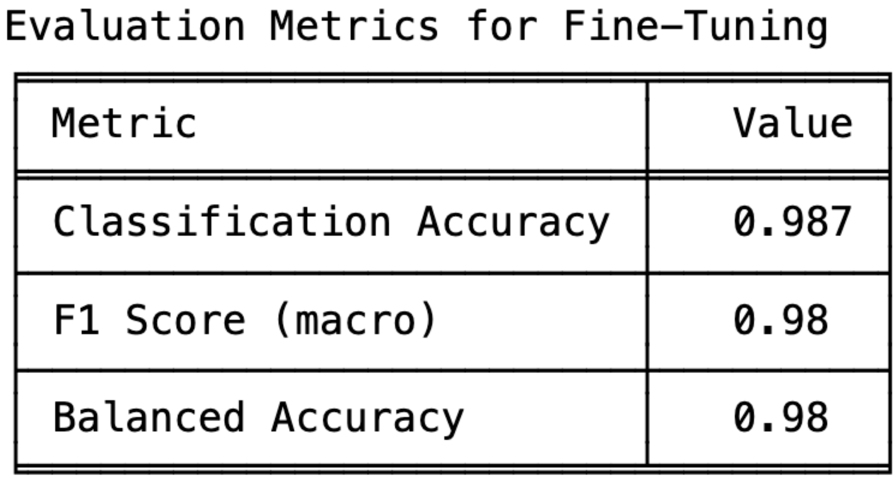


Fig. 5

Cost Plot for Fine-Tuning Transfer Learning



A) Evaluation Metrics for Fine-Tuning

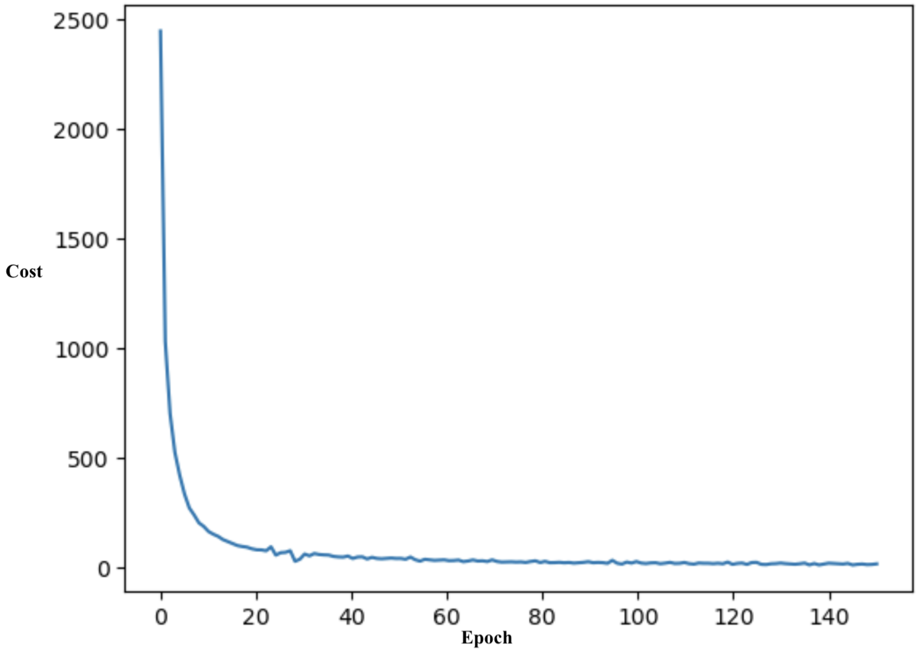


Evaluation Metrics for Fine-Tuning

Metric	Value
Classification Accuracy	0.987
F1 Score (macro)	0.98
Balanced Accuracy	0.98

Fig. 6

A) Cost Plot for Feature Extraction Transfer Learning



B) Evaluation Metrics for Feature Extraction

Evaluation Metrics for Feature Extraction

Metric	Value
Classification Accuracy	0.995
F1 Score (macro)	0.99
Balanced Accuracy	0.99