Slide 1: Track 2 CNNs & Transfer Learning

"Hello everyone, and welcome to my presentation on 'CNNs and Transfer Learning,' specifically comparing these two approaches for object recognition using the CIFAR-10 dataset. This presentation will cover key aspects of the project, including the architectures of two models we've implemented: a custom Convolutional Neural Network (CNN) and a Transfer Learning model using a pre-trained base model. Both of these models will be evaluated based on their performance in classifying images from CIFAR-10, a standard dataset for image classification tasks. By the end of this session, we will have a clear understanding of how well these two techniques perform, and the advantages and challenges associated with each."

Slide 2: Agenda & Introduction

"Let's begin by outlining the structure of the presentation. First, I'll provide an introduction to the project, explaining the objective and goals of this research. Following that, I will explain the data preparation process, including how the dataset was split and the rationale behind validation. Next, we'll dive into two different models: Model 1, which is a custom-built CNN architecture, and Model 2, which uses Transfer Learning with a pre-trained model. I'll then compare the performance of both models based on metrics like accuracy, precision, and recall, followed by a discussion on how Track 2 (Deep Learning) compares with other tracks like classical machine learning and advanced machine learning techniques. Finally, I'll conclude the presentation by sharing the lessons learned and the key takeaways from this project, including possible directions for future work."

Slide 3: Data Preparation & Validation Rationale

"To start off with the data, I used the CIFAR-10 dataset, which consists of 60,000 32x32 color images from 10 different classes. The original split of the dataset is 50,000 training images and 10,000 test images. However, for the purpose of this project, I divided the dataset into 70% training, 15% validation, and 15% test, amounting to 35,000 images for training, 7,500 images for validation, and 7,500 images for testing. The reason I used this specific split is to ensure that we have a sufficient amount of data for training while also having enough validation and test samples to assess model performance fairly.

"Regarding data preprocessing, I normalized the images by scaling pixel values between 0 and 1. This is important because it helps in faster convergence during training by ensuring that the input values are within a similar range. Additionally, I applied data augmentation techniques to the training set, such as random rotations, horizontal flips, and zooming, to introduce more variability into the data and help the model generalize better. This is especially important when working with small datasets like CIFAR-10, where models tend to overfit easily.

"One crucial point in this process was the decision to use a validation set. A validation set is essential for tuning hyperparameters and preventing overfitting. Without a validation set, we could risk training the model so well on the training data that it doesn't perform effectively on unseen data, which is a common problem known as overfitting. The validation set helps in evaluating the model at each step of training and selecting the best set of hyperparameters."

Slide 4: Model 1 - Custom CNN Architecture

"Now let's dive into Model 1, which is based on a custom Convolutional Neural Network (CNN) architecture. CNNs are particularly suited for image classification tasks due to their ability to automatically detect spatial hierarchies of patterns in images, such as edges, textures, and object parts. The architecture I designed for this model consists of several layers, each serving a specific purpose to extract features from the images and then classify them.

"The input to the network is a 32x32x3 image (CIFAR-10 images are small, with 32x32 pixel resolution and 3 color channels). The network consists of two convolutional blocks. Each convolutional block includes a 2D convolutional layer followed by a max-pooling layer. The convolutional layers help extract low-level features, such as edges and textures, from the input images. Max-pooling layers are used to reduce the spatial dimensions of the image, which not only reduces computational cost but also helps prevent overfitting by simplifying the learned features.

"After the convolutional blocks, I added a dropout layer with a rate of 0.25. This helps to prevent overfitting by randomly setting a fraction of input units to 0 during training, which forces the network to generalize better. Following this, we have the classifier part of the network, which flattens the output from the convolutional blocks and passes it through dense layers with ReLU activation. The final layer of the classifier consists of 10 units, each corresponding to a class in the CIFAR-10 dataset, with softmax activation to generate probability scores for each class.

"Training the model involves several key choices. I used the Adam optimizer, which is an adaptive optimizer that adjusts the learning rate during training for more efficient optimization. The loss function used is categorical cross-entropy, which is appropriate for multi-class classification problems. I set the learning rate to 0.001 and used a batch size of 32 or 64, with early stopping on the validation set to avoid over-training."

Slide 5: Model 2 - Transfer Learning

"Moving on to Model 2, which leverages Transfer Learning. Transfer learning is a technique where we use a model pre-trained on a large dataset (like ImageNet) and adapt it to our own task with relatively small data. The intuition behind this approach is that the lower layers of a pre-trained model have learned general features, such as edges and textures, that can be useful for many image classification tasks, while the upper layers can be fine-tuned for specific tasks.

"For this model, I used VGG16, a well-established architecture that has been pre-trained on ImageNet, a much larger dataset. Since the CIFAR-10 images are only 32x32 pixels, while VGG16 expects images of size 224x224, I upscaled the CIFAR-10 images to 224x224. While this does introduce a slight loss of information, it was necessary to make the data compatible with VGG16's input requirements.

"After using the pre-trained VGG16 model, I removed the top layers (the classifier) and froze the weights of the convolutional layers. Freezing the layers means that these weights will not be updated during training, and only the newly added classifier will be trained. This is a standard practice in transfer learning since the lower layers have already learned useful features and don't need to be re-trained.

"The final step in the transfer learning pipeline is fine-tuning. After training the new classifier, I unfreezed the last few convolutional layers of VGG16 and trained them with a very low learning rate. This allows the model to subtly adjust the pre-trained features to better suit the CIFAR-10 dataset."

Slide 6: Performance Metrics & Results

"Now, let's take a look at the performance results of both models. Model 1, the custom CNN, achieved a test accuracy of around 80%. The precision (macro) for the custom CNN was 0.81, and recall (macro) was 0.80, which are respectable scores, but not optimal. On the other hand, Model 2, the transfer learning model, achieved a much higher test accuracy of around 88%. The precision (macro) was 0.89, and recall (macro) was 0.88, indicating that the transfer learning model was much better at classifying the CIFAR-10 images and minimizing false positives.

"The chart on the right shows a comparison of both models' test accuracy during fast training. As we can see, the transfer learning model has significantly better accuracy. The advantage of transfer learning here is clear: by leveraging knowledge from the pre-trained model, we were able to improve the performance substantially without needing as much data or training time."

Slide 7: Live Demonstration

"Let's move to the live demonstration. I will now load the saved transfer learning model and select a few sample images from the CIFAR-10 test set. I'll run the model on these images and show the predictions, including the top 3 predicted classes along with their confidence scores. This will give you a sense of how well the model generalizes to new, unseen data. During this demonstration, we will also see how confident the model is in its predictions by displaying the confidence scores."

Slide 8: Comparative Discussion

"Let's now take a step back and discuss the comparative advantages and challenges of Track 2 (Deep Learning). One of the main strengths of deep learning is its ability to achieve state-of-the-art performance on image classification tasks. CNNs are capable of automatically extracting features from raw data, eliminating the need for manual feature engineering. Furthermore, deep learning models like CNNs are highly flexible and can be scaled easily for more complex tasks.

"However, deep learning also has its trade-offs. The most notable challenge is that it requires large amounts of data to train effectively. This is mitigated by techniques like data augmentation and transfer learning, but the demand for data is still much higher than traditional machine learning methods. Additionally, deep learning models are computationally expensive, requiring powerful GPUs for training, and they are often considered 'black boxes,' making them harder to interpret compared to simpler models like Support Vector Machines (SVM) used in Track 1.

"When comparing Track 2 with Track 1 (Classical ML), it's clear that deep learning models outperform traditional methods in terms of accuracy, but they come with higher resource requirements. When comparing to Track 3 (Advanced ML), Track 2 methods are well-established and require less hyperparameter tuning. However, Track 3 techniques like Neural Architecture Search might achieve better performance with sufficient computational resources."

Slide 9: Conclusions & Lessons Learned

"To summarize, deep learning, particularly CNNs with transfer learning, proved to be highly effective for object recognition on the CIFAR-10 dataset. By leveraging techniques like data augmentation and dropout, we were able to prevent overfitting and achieve good performance even with a relatively small dataset. Transfer learning provided a significant performance boost by utilizing pre-existing knowledge from larger datasets like ImageNet.

"However, there were also several lessons learned during the process. One of the key takeaways was the importance of having a rigorous train/validation/test split to ensure reliable results. Additionally, we found that the model's performance is highly sensitive to hyperparameters and architectural choices, which require careful tuning. Choosing the right optimizer and learning rate schedule is also critical for achieving optimal performance."

Slide 10: References

"Finally, here are the references that guided my work. These papers cover various aspects of deep learning and transfer learning, and I encourage anyone interested in diving deeper into the topic to explore them. They provide a wealth of knowledge on model architectures, fine-tuning strategies, and the theoretical foundations of these techniques."