Tony Ho Capstone Proposal

Domain Background

I'm choosing the Inventory Monitoring at Distributions Centers project. In distribution centers, there are robots that hold bins containing a varying number of items. However, there may be errors resulting in the expected number of items in the bin not matching the actual number. Detecting these errors early can result in substantial benefits when applied to a large-scale operation, such as in an Amazon Distribution Center. The actual number of items can be identified by evaluating an image of the bin, taken from a top-down view. The goal of this project is to train a deep learning model that can classify an image of the bin's contents as a bin with the correct number of items. For example, if the image showed only two distinct items, then, it should be classified as a two-item bin.

This problem has previously been tackled using SVMs and CNNs, particularly in the paper Amazon Inventory Reconciliation with AI (Bertorello et. al, 2018). The researchers showed CNNs far outperformed SVM on classification but were ultimately only able to reach ~56% validation accuracy using ResNet34.

This project is an example of an industrial and practical use case of computer vision, which greatly interests me since I'm a master's student with a focus in the computer vision field. I will initially try to replicate these results using the AWS Sagemaker workflow and look for ways to improve my model.

Problem Statement

The challenge is to build a deep-learning model that can correctly classify the images as accurately as possible. Amazon has provided a dataset with over 500,000 images of bins containing 0-5 items. The dataset contains some images that are challenging even for humans to classify. There's also typically a clear tape that occludes some of the items in the bin, making it even more difficult to detect and deters feature engineering.

Solution Statement

I should at least be able to achieve 50% accuracy using the ResNet34 model. Leveraging AWS's compute resources, I may investigate using a Vision Transformer (ViT) model and

compare its performance. The pretrained ViT should be adept at detecting objects and since transformers are capable of applying that knowledge to a classification task, I'm hoping for good results.
Using Amazon Sagemaker, I'll download the full dataset and upload it to my S3 bucket. In a separate Python script, I'll setup the ResNet34 model and training loop based on the implementation from my previous projects in this course. I'll adapt the Dataloader to implement the horizontal image augmentation and image resizing. The paper uses the Adam optimizer with learning rate of 0.001 and 0 weight decay.

Datasets and Inputs

The dataset has been provided and features 535,234 images and the same number of json files that contain metadata on the corresponding image. Each image will have a top-down view of a bin that will contain 0-5 items. The previously mentioned paper resized the images to 224x244, used 324,000 images and normalized the pixel data. Their train-test-split was 70% for training, 20% for validation and 10% for testing. Each image was also horizontally flipped to augment the dataset. Since AWS has powerful compute resources, I'll consider increasing the image size and using the full dataset if I'm able to replicate the expected results.

Benchmark Model

Although a trained model itself isn't provided, a similar model should be easily trained. A ResNet34 model using the same image augmentations is expected to perform above 50% accuracy.

Evaluation Metric

To be consistent with the paper, I will use the accuracy metric. This metric counts the number of correctly classified images and divides it by the total number of predictions to produce the accuracy percentage.

Project Design

Using Amazon Sagemaker, I'll download the full dataset and upload it to my S3 bucket. I'll use 20%, or about 100,000 images for the purpose establishing parity with the model outlined in

paper Amazon Inventory Reconciliation with AI. If performance is significantly less than expected, such as 45% accuracy as opposed to ~50%, then more training samples will be included, up to 300,000. Like previous projects, a separate python script just for loading the Resnet34 model without pretrained weights and performing the training loop. In a sagemaker notebook, I'll create the estimator using the sagemaker PyTorch class with the previously mentioned python script as the entry point. The hyperparameters that were used in the paper are known and will be directly plugged into the optimizer. Training such as model is as simple as creating an estimator object with the python script as the entry point and then calling the fit function.

After evaluating the model for parity, I'll perform training on the full dataset, apply typical image augmentations, and evaluate performance from there. Such augmentations include random horizontal flip and image resizing. Depending on the training time, I may consider using the AdamW optimizer instead and perform hyperparameter optimization, then training again with the full dataset.

Finally, I'll investigate using the Vision Transformer model and fine-tuning the pretrained model to make predictions for this specific dataset. Using the base 32 variant, I'll first do hyperparameter optimization in the same way as previously outlined, using the AdamW optimizer. Then I'll train the estimator on the entire dataset with the same image augmentations outlined previously and evaluate the results.

References

Rodriguez Bertorello, P., Sripada, S., & Dendumrongsup, N. (2018). Amazon Inventory Reconciliation Using AI. *Available at SSRN 3311007*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.