Sheet Defect Detection from Videos

Sarosij Bose, Department of Computer Science and Engineering, University of Calcutta.

Problem Statement and Approach

- Problem Statement: Detect defects from printed sheets from videos.
- Approach:-

1. Use a pre-trained ImageNet model (here, we use **InceptionV3**).
2. Next, apply Transfer Learning by fine-tuning it on a quality control image dataset by adding a few fully connected layers.
3. Get the training and testing accuracies.
4. The dataset used here is the **Severstal Steel defect detection** dataset.
5. Finally, extract a key frame from the provided video (here, we extract the centre frame.) and feed it to the model.

Updated Approach

- Approach:-

1. Use a pre-trained ImageNet model (here, we use InceptionV3).
2. Next, apply Transfer Learning by fine-tuning it on the frames of the provided videos by adding a few fully connected layers.
3. Train the binary classifier and get the training and testing accuracies.
4. The dataset used here is the **frames obtained from the provided videos.**
5. Finally, **feed the test split of the frames** to the model instead of a single selected frame earlier.

Components of the Pipeline

- The Pipeline consists of the following modules:-

1. Preprocessing/Frame extraction module.
2. Network loading and training module.
3. Inference module.
4. Camera Interface for direct video feed input.(To be incorporated)

- Pipeline time taken:-

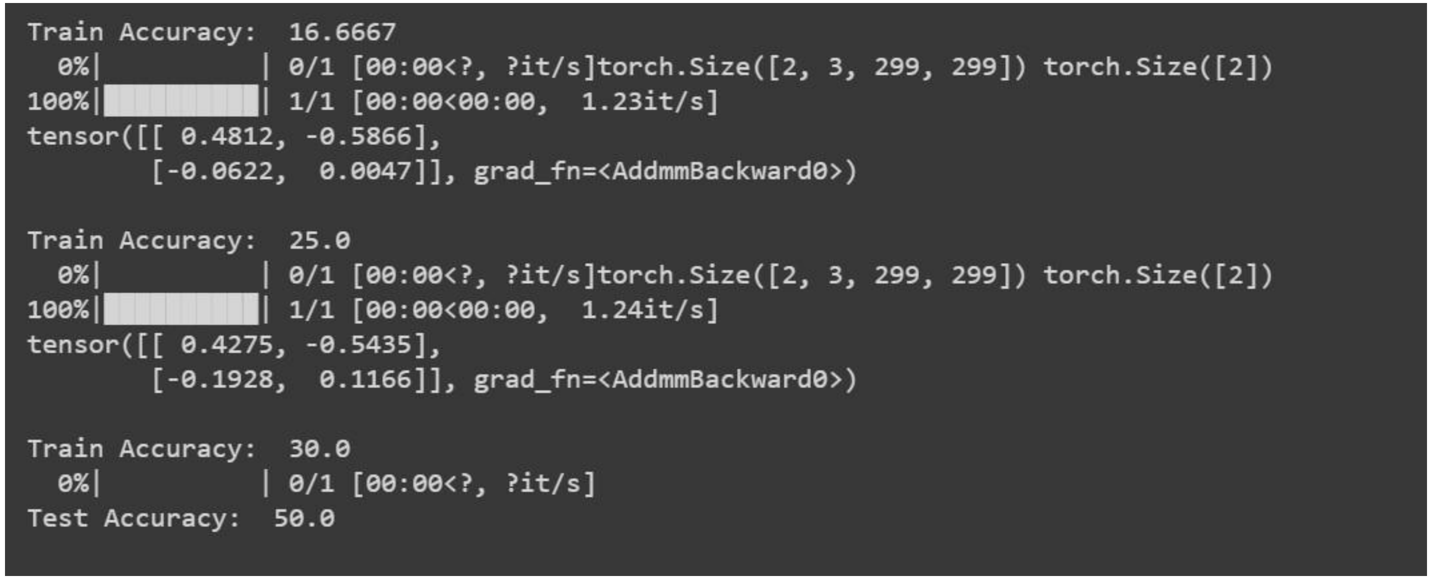
End to end time taken: 8 mins.

Frame extraction time: 1.2 seconds.

Inference time: 0.8 seconds

Result Analysis

A suspected reason maybe the **domain/distribution gap** which the model may have been facing. On training the model with the frames of the video however, the accuracy increased two-fold from **28% to 50%.**



```
Train Accuracy:  16.6667
  0%|          | 0/1 [00:00<?, ?it/s]torch.Size([2, 3, 299, 299]) torch.Size([2])
100%|██████████| 1/1 [00:00<00:00,  1.23it/s]
tensor([[ 0.4812, -0.5866],
        [-0.0622,  0.0047]], grad_fn=<AddmmBackward0>)

Train Accuracy:  25.0
  0%|          | 0/1 [00:00<?, ?it/s]torch.Size([2, 3, 299, 299]) torch.Size([2])
100%|██████████| 1/1 [00:00<00:00,  1.24it/s]
tensor([[ 0.4275, -0.5435],
        [-0.1928,  0.1166]], grad_fn=<AddmmBackward0>)

Train Accuracy:  30.0
  0%|          | 0/1 [00:00<?, ?it/s]
Test Accuracy:  50.0
```

Reasoning

- Some of the reasons why the above approach was taken is listed below:-
- Since no pre-trained spatiotemporal models for defect detection are available yet, a image based approach was taken here. While it is known that ImageNet pre-trained models have their limitations for Video classification, it can still perform quite well with proper initialization and architecture.
- Further, the problem is exacerbated with the unavailability of any public video based dataset for defect detection. Hence, the usage of image dataset here.

Scope for Improvement

These are some of the areas where the model can be improved:-

1. Instead of just extracting the centre frame, we can use entropy to detect whether there is any change in the HOG of the frame and select frames accordingly.
2. Try to use a LSTM based approach – the best approach to image based pre-training for classifying videos.
3. Finally, loop the images to form a “still video” and use these for training.

Resources

- Link to Code:- (In Google Colab)

https://colab.research.google.com/drive/1YhPFxivk19uJ0AQdhZUsYb-Mu8y7pukI?usp=sharing

- Link to Dataset:- (Kaggle, used earlier)

https://www.kaggle.com/c/severstal-steel-defect-detection

Thank You!