Deep Learning for Human Action Recognition

Nishq Ravindranath

1 Problem Definition and Analysis

The problem here is to predict human actions from the dataset of images provided. It falls under the area of Human action recognition (HAR). This is a Multi-task learning problem, with two tasks that need to be predicted, one is the action class, and the other is the action, which is a subset of specific action classes. There are 21 types of actions belonging to 5 action classes.

Exploratory Data Analysis

The data was first loaded in, and a sample of the images were visualized, and the differences in their dimensions was checked. The maximum and minimum values from the dimensions of the images were then calculated to further guide what sort of technique should be used for the data loader. It was found that the smallest x and y values of the images was 200, and the largest was 961. This is a very large variance in dimensions, and for our model we want them all to have the same dimensions, while also retaining the characteristics of each image that makes them have their associated label.

The dataset was then analysed in terms of its class distribution. This was done for both action_class, and the action within them. For action_class, it was found that there is a class imbalance in the dataset, with "other_activity" having as many as 3 times the number of images as "playing musical instrument".
The classes "domestic work", "interacting with animal" and "using_comm_device" are nearly evenly repre sented in this dataset. The same was done for action within each action_class, and it was found that “playing_musical_instrument" and "interacting_with_animal" are well balanced.
The other 3 action classes have a lot of imbalances with their sub actions. The EDA guided the choice of the performance metric for each class and the subsequent evaluation framework.

2 Evaluation Framework

The performance metric chosen for each class is as follows:

- action_class: **Accuracy**. Even though there is an imbalance in class distribution, this is somethin g we could accommodate for with data augmentation. Moreover, if we consider the distribution i gnoring "other_activity", it is not too imbalanced. "other_activity" is a generic action_class for a rguably a more diverse set of actions than the other action_classes. Thus, the standard accuracy metric should be good enough.

- action: **Top 5 Categorical Accuracy**. Chosen because there are 21 classes in total, with a wide variety of different actions, that can sometimes be interpreted differently. For example, "cooking ", "washing dishes", and "cutting vegetables" are 3 activites that tend to have the person doing th em in a similar posture. Our deep learning model may interpret this posture to be the primary det erminant of classifying these. Moreover, sometimes there is overlap in the images themselves. F or example, a person may be cutting vegetables, but have the stove on with food cooking and dis hes in the sink all simultaneously. This can confuse the model, so having a "top 5" seems more a ppropriate.

Hold-out validation was used to evaluate the different models, with the two main performance metrics’ validation score assessed for each class. Tensorboard was used to monitor the performance metric at each epoch, and early stopping with a patience ranging from 5-10 (depending on the parameter changes) was employed to stop the model if the model's performance started to plateau. Finally, the results of the models were plotted to compare and evaluate the changes of the different parameters.

3 Approach & Justifications

Since the images were of different sizes, they were all first resized to a unified 224x224. This size was chosen as it is a common image size to input for images that convey more complex data. Furthermore, upsizing smaller images to a higher resolution might make them lose some details due to fuzziness. Higher resolution images will also require a lot more computational time for a convolutional neural network.

The EDA made it clear that there was a scarcity of available data to train the model. Given the number of different classes for both tasks, and the fact that this is an image classification task, data augmentation was used to generate more images. The data augmentations chosen from visualization and exploration were:

- Horizontal flipping
- Rotation (20 degrees, clockwise and anti-clockwise)
- Image cropping
- Gamma adjustment (brightening and darkening)
- Adding random noise

Combinations of each of these augmentations with each other were applied to the images, producing an additional 32 augmented images per image. The augmentation was added into the data generator, which loads the data into the model. Furthermore, since the problem here is of multi-task learning, the data generator was also given the ability to load images with two classes.

An evaluation pipeline was then setup, which included getting the metrics as discussed in the Evaluation Framework section. A baseline model was then chosen – ResNet-34, the first model using the residual block framework on ImageNet. This model was selected to test whether the original architecture could generalize well to other image recognition problems, and with a much smaller and limited set of data.

4 Experiments & Tuning

Baseline Model Performance

The baseline model performed poorly on the dataset, with action_class underfitting, and action overfitting. The model also seemed to plateau very quickly. It was theorized that this may be due to the momentum being too high (0.9) causing the model to have difficulty estimating the unknown target function. Decreasing the momentum was then tried.

Experimentation – Momentum

First, a random search was done, and the momentum was halved to 0.4. A semi-grid search was then performed based on the performance evaluations. This resulted in a slight performance increase, although not nearly as much as was expected. The momentum was thus further reduced to 0.1, which resulted in a performance worse than the baseline model. The model was still generally underfitting for action_class, and underperforming for action. It was thus decided to experiment with increasing the complexity of the model.

Experimentation – Increasing Model Complexity

The model's complexity was increased to address the underfitting concerns. This was done both by adding more hidden layers to the MLP, as well as adding more ResNet blocks. Neither resulted in any kind of performance improvement – both were worse.

Experimentation – Changing Optimizers

Two additional optimizers were then tested: Adam and RMSprop. Both are **adaptive** optimizers that don't need a learning rate schedule specified. Adam is also commonly used for image classification. Both optimizers increased performance on both the training data and validation data, however they were both overfitting.

Experimentation – Regularization

For regularization, first a reduction in the batch size was tested, since it tends to have a regularizing effect. However, there was no considerable difference in performance, and it only increased the time for training. L1 and L2 regularization were then tested, with the lambda parameter set to 0.01 as a default at first. It was noted from this that the model was severely underperforming, therefore the lambda was reduced ten-fold and then tuned from there on using half-step values between 0.01 and 0.001. it was determined that an L2 regularization with 0.001 lambda was best. The overall performance of the model seemed to be hitting a ceiling at this point, thus the focus was shifted to trying a different activation function for the ResNet blocks.

Experimentation – Activation Function

Leaky ReLU was tried, which has a small slope for negative values instead of a flat slop like ReLU. It is commonly used for image classification tasks and tends to have better performance. However, there was no improvement in performance, in fact it did slightly worse.

5 Ultimate Judgment, Analysis & Limitations

The final model chosen was the custom tuned model with momentum = 0.4, same complexity as the baseline model, Adam as an optimizer, and L2 regularization with lambda = 0.001. This model had the best performance, with an action_class accuracy of 0.65, and a top 5 action accuracy of 0.83. This is not a great performance, however, given the limited amount of data and the various class imbalances, it can be considered decent.

ResNet-34 might not be suitable for multi-task learning on this dataset, unless of course it can be tested with more data, and with no class imbalances as was present in this dataset. Given more time, I believe there is more experimentation that can be done with the multi-task learning architecture. We can have architectures where both tasks share only the first few layers worth of parameters, and then branch off separately.

Moreover, dropout regularization can also be tested with an increased model complexity with Adam as an optimizer. Furthermore, experiments can also be done on changing the padding, number of filters, stride length, and type of pooling in the convolutional layers.

6 References:

- Parsa, Behnoosh, and Ashis G. Banerjee. "A Multi-Task Learning Approach for Human Activity Segmentation and Ergonomics Risk Assessment." *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, doi:10.1109/wacv48630.2021.00240.
- Siyal, Ahsan Raza, et al. "Still Image-Based Human Activity Recognition with DEEP Representations and Residual Learning." *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 5, 2020, doi:10.14569/ijacsa.2020.0110561.
- He, Kaiming, et al. "Deep Residual Learning for Image Recognition." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, doi:10.1109/cvpr.2016.90.
- Thakur, Ayush. "How to Handle Images of Different Sizes in a Convolutional Neural Network." *W&B*, 10 Aug. 2020, wandb.ai/ayush-thakur/dl-question-bank/reports/How-to-Handle-Images-of-Different-Sizes-in-a-Convolutional-Neural-Network--VmlldzoyMDk3NzQ.
- Giordano, Davide. "7 Tips to Choose the Best Optimizer." *Medium*, Towards Data Science, 26 July 2020, towardsdatascience.com/7-tips-to-choose-the-best-optimizer-47bb9c1219e.