Machine Learning Engineer Nanodegree

Capstone Proposal

Dhirendra Kumar
December 10, 2018

Proposal

The Goal of this project is to develop a model capable of classifying mixed patterns of proteins in microscope images. However, unlike most image labeling tasks, where binary or multiclass labeling is considered, in this competition each image can have multiple labels.

Here all image samples are represented by four filters (stored as individual files), **the protein of interest (green)** plus three cellular landmarks: **nucleus (blue)**, **microtubules (red)**, **endoplasmic reticulum (yellow)**. Therefore, An additional challenge is 4-channel input to the model (RGBY), which is different from ones used in most of pretrained models (RGB input).

Images we will be using for this are generated with the help of microscope and at a far greater pace than what can be manually evaluated. Therefore, the need is greater than ever for automating biomedical image analysis to accelerate the understanding of human cells and disease.

Domain Background

Proteins are "the doers" in the human cell, executing many functions that together enable life. Historically, classification of proteins has been limited to single patterns in one or a few cell types, but in order to fully understand the complexity of the human cell, models must classify mixed patterns across a range of different human cells.

Images visualizing proteins in cells are commonly used for biomedical research, and these cells could hold the key for the next breakthrough in medicine.

Problem Statement

Classification of proteins has been limited to single patterns in one or a few cell types, but in order to fully understand the complexity of the human cell, models must classify mixed patterns across a range of different human cells. And automating the biomedical image analysis will accelerate the understanding of human cells and disease.

Datasets and Inputs

The data format is two-fold:

The bulk of the data is in the images (scaled set of 512x512 PNG) - **train.zip** and **test.zip**. Within each of these is a folder containing four files per sample. Each file represents a different filter on the subcellular protein patterns represented by the sample.

The labels for **train.zip** are provided for each sample in **train.csv.** And the labels for **test.zip** are not provided because the prediction of the developed model will be compared with the actual labels after the competition ends. So, here images from the **train.zip** will be used for the development of the model and the images from **test.zip** will be used for evaluation of the model.

There are in total 28 different **protein organelle localization labels** present in the dataset. The dataset is acquired in a highly standardized way using one imaging modality (confocal microscopy). However, the dataset comprises 27 different cell types of highly different morphology, which affect the protein patterns of the different organelles. All image samples are represented by four filters (stored as individual files), **the protein of interest (green)** plus three cellular landmarks: **nucleus (blue)**, **microtubules (red)**, **endoplasmic reticulum (yellow)**. The green filter should hence be used to predict the label, and the other filters are used as references.

The data for this project is provided by Kaggle and Human Protein Atlas, should be found here

Solution Statement

The goal of this competition is classification of mixed protein patterns. That can be achieved by using CNN models. However, unlike most image labeling tasks, where binary or multiclass labeling is considered, in this competition each image can have multiple labels. Which can be solved by using a function similar to sigmoid function to get more than one labels for sample images.

An additional challenge is 4-channel input to the model (RGBY), which is different from ones used in most of pretrained models (RGB input), which can be solved by just dropping on color chanel from the images. I will be using some other libraries, model, tools and techniques if necessary.

Benchmark Model

Since the proposed project is actually a kaggle competition, the benchmark will be the best kaggle score for test set on **Public or Private Leaderboard** depending on weather I submit the capstone project before or after the competition ends, because kaggle will use only **29%** of the **test data** to calculate the **F1 score** before the deadline and the F1 score on full test data will be calculated and revealed when the competition ends.

So, I will be using different CNN models and other Computer Vision techniques and making comparison between their F1 scores to develop the best model. And my goal will be get as high as I could in leaderboard of this competition.

Evaluation Metrics

The official evaluation of the developed model is done by kaggle using **F1 score**. **F1 score** of a model is calculated by the the **Harmonic Mean** of **precision** and **recall**. where an F_1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

Where precision is the ratio of the number of true positive and the all positive labels predicted by proposed model, And recall is **is the number of correct positive results divided by the number of all positive results returned by the classifier, and *r* is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).**

To understand it better let's say we have total **m+n** labels for some dataset, in which labels for **m** samples are **positive** and for the rest of **n** is **negative**, and when we applied the proposed model to these samples, it predicted **positive** for **P** samples and prediction was correct for **Q (Q<=P)** samples, then precision is **Q/P** and recall is **Q/m.**

Project Design

The summary of the the workflow for approaching the solution will be something similar like this:

Importing necessary libraries → Loading and reading our data → Plotting several images to develop some sense about the data → splitting the training dataset in training and validation datasets → Loading or creating some computer vision models → Performing data augmentation → Defining the loss function → Training the model → validating the model with validation data → Tuning the model parameters → Validating the model with Test Time Augmentation to get the better predictions → Training the developed model with whole training

data (including validation data) → making prediction for test data → submitting the results to get the final F1 score of the model on test data.

References

1. Human Protein Atlas Image Classification.
   https://www.kaggle.com/c/human-protein-atlas-image-classification
2. F1 score.
   https://en.wikipedia.org/wiki/F1_score
3. kaggle kernel
   https://www.kaggle.com/iafoss/pretrained-resnet34-with-rgby-0-460-public-lb