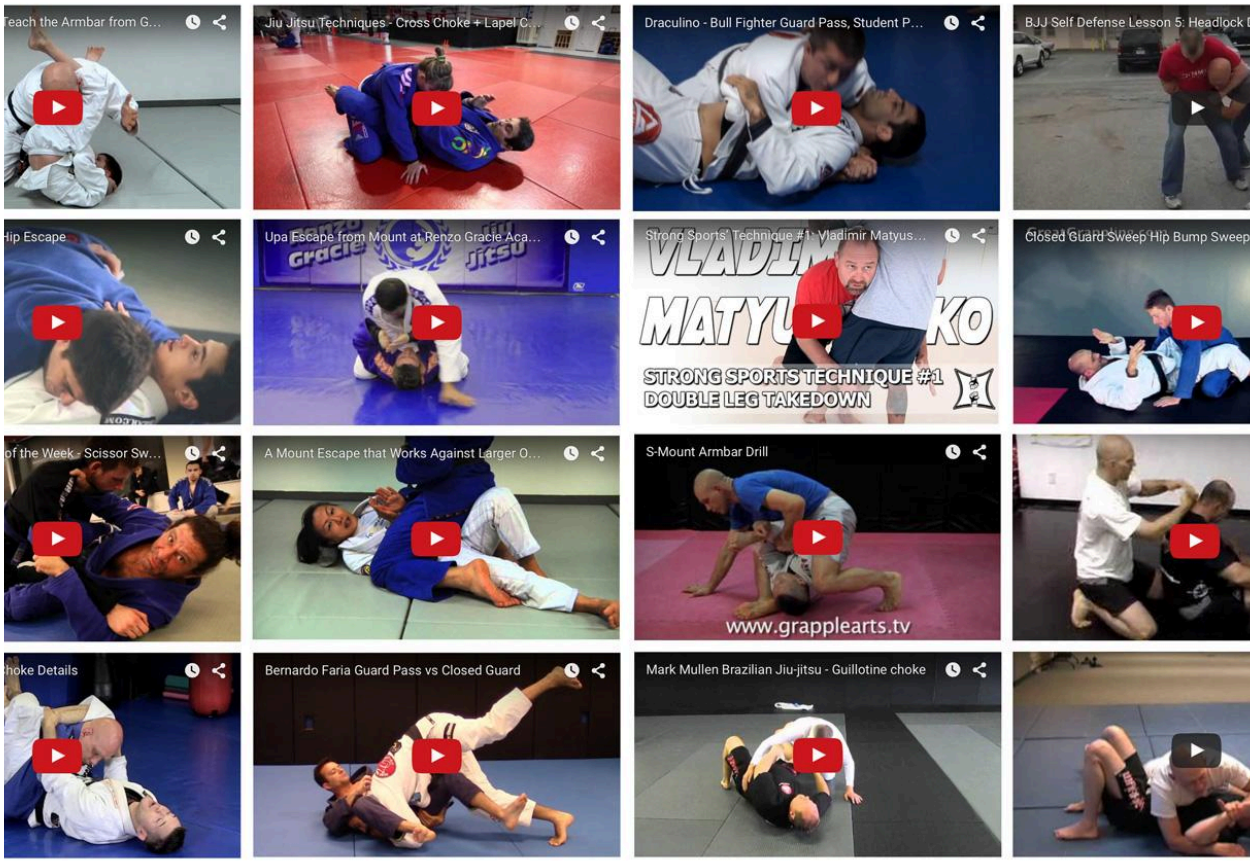
Project SAIBANKAN
Akhil Singh Rana and Dave Aitel

11/30/2017



Introduction

Test our code with your images: http://second-150623.appspot.com/

This project envisions a world where cameras replace Brazilian Jiu-Jitsu (BJJ) judges using computer vision algorithms. Typical BJJ points are scored based on different labeled “positions” and the transitions between them, with some positions being considered more dominant and hence worth more points. So the first step towards an automatic judge is a

computer vision system that can look at a BJJ match and say what position is recognized, and who is “on top”.

SAIBANKAN uses Google cloud services and their hosted Tensorflow framework entirely, to avoid any scalability issues down the line. Google’s Inception-v3 deep learning model was fine-tuned and trained on Google ml-engine on the Images collected and scraped from our web API, which was written using the Python Flask framework and deployed on Google app-engine. This report will go through the basic steps completed and images from some of the results which we achieved. We will discuss the data collection pipeline and services used.

Obviously part of this effort was designed to build a pipeline for similar projects, and discover any pitfalls which might affect such future projects.

Procedure

- **WEB API and Data Collection Pipeline**-
    - Framework used Python flask and templates were developed with basic HTML5.
    - Google’s Firebase was used for easy one-click authentication, support for facebook, gmail and email login.
    - BJJ Images were scraped from video sources (YouTube), and post-processed by a Python script that extracted interesting frames from the video.
    - Images were then directly stored to the cloud storage bucket.
    - Cloud datastore API was used for maintaining the database (including user details, Images, and videos already processed).
    - Google cloud’s PUB/SUB is used for handling video downloads, image extraction and storage run in the background without interfering with users in the frontend.
- **Image Labeling**
    - Stored Images are then shown to logged-in users. Any admin user gets access and control over every part of the process from video downloading to labeling.
    - Labelled Images are moved to a particular Storage bucket.

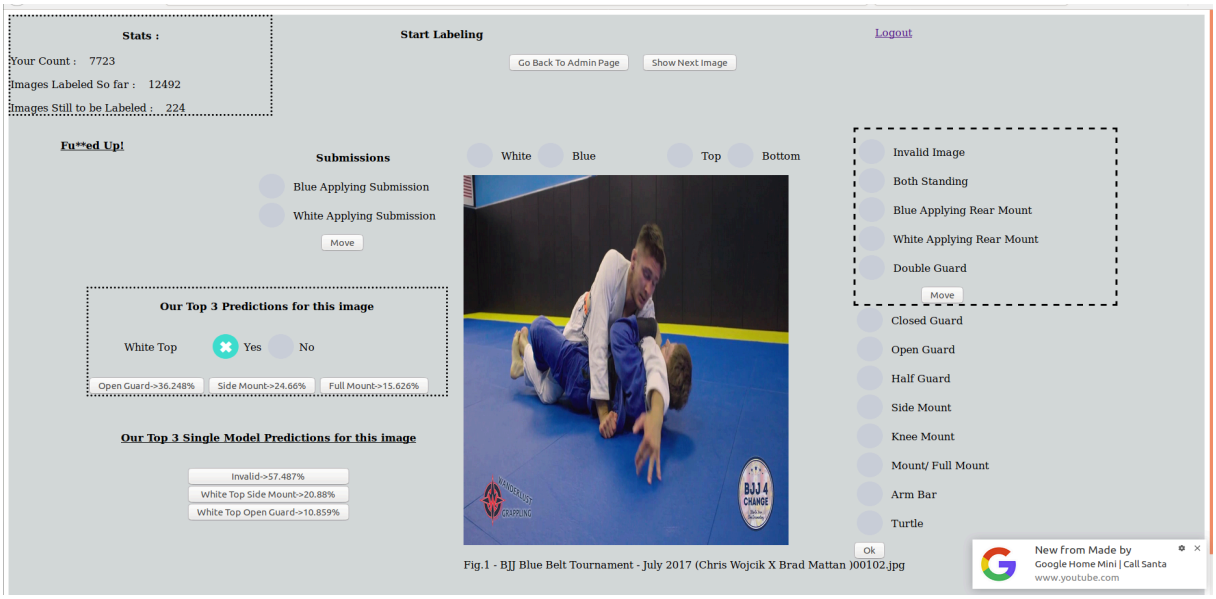


Fig.1 - BJJ Blue Belt Tournament - July 2017 (Chris Wojcik X Brad Mattan)00102.jpg

The above screenshot shows the labeling UI. This UI is more complicated than originally envisioned but on the left is a simplified interface which allows the user to select whether the ML is “correct” and avoid the manual labeling process. It also allowed the researchers to get an intuitive view into how well the ML process was assigning confidence to its guesses.

For some positions, such as “standing”, there is no “TOP” or “BOTTOM”. Likewise, all videos were selected to have participants with a white gi and a blue gi.

- **DeepLearning Part**
    - These collected images are then used to train the neural network
    - Google’s Inception v3 is used for classification.
    - Cloud dataflow API is used for parallel and fast data processing to convert images to tfrecord format.
    - Then this trained model is used for future frame prediction and help users in labeling Images by preselecting or showing the labels which our neural network thinks the image belongs to. As also can be seen from the above image.
    - We made two models
        - 1st one has separate neural networks one predicting color of players and another one predicting the position.

        - 2nd model is a single model which predicts both color and position of player. Note at this time we had quite a few images in the dataset in order for this to be possible.

Model Accuracy

- Double model with one for position and one for “top color”
    - “Back control” had to be handled and labeled differently from, for example, mount or half-guard or closed-guard as “back control” can have the dominant person on the bottom or top
    - Model trained for 1400 Iterations on September 2017 with following dataset labels and the respective number of images
        -

Label	**No. of Images**
Closed Guard	1188
Open Guard	1540
Half Guard	719
Side Mount	341
Knee Mount	67
Full Mount	676
Turtle	286
Arm Bar	0
Double Guard	73
Blue applying submission	67
White Applying submission	49
Blue Applying Rear Mount	93
White Applying Rear Mount	223

    - Accuracy achieved by this model was 77% on the evaluation test, which is quite interesting based on how badly distributed the dataset is.

- Single Model
    - The dataset remained same as above with just changing the network to now predict both color and positions.
    - Accuracy achieved was 69%, which is still a reasonably high result.

Lessons Learned

BJJ positions are often judged, by a human, with a lot of context information from previous positions. For example, the position “half-guard” is often followed by mount, side control, or a sweep (transition from bottom to top). In many cases, the limbs of a particular participant will be occluded from view from the camera (which cannot move, unlike the judge).

Other discovered issues are that the labels are by definition fuzzy: Half guard and open guard are a continuous spectrum, for example. In other words, two humans may disagree on what a particular position is and where the boundaries between them are. “S-Mount” is essentially equivalent to “Full Mount” but looks very different. Likewise, is “Deep Half” its own position or should it be labeled as “Half Guard”? These kinds of questions can be subtle and difficult as the going philosophy in BJJ scoring may not be accurate to reality.

This is especially true for many positions which cannot be “labelled” as anything in particular. In some cases, a “triangle attempt” can come from closed guard, but the transition positions to it are not closed guard, nor is the person on top necessarily in the dominant position. Because there are essentially infinite “submission” positions where one participant is having a limb almost broken or being choked unconscious, it can be difficult to label them as anything other than “Blue is attempting a submission” or in many cases as “invalid”.

That said, **covering the basic positions and who is on top allows for gathering extremely useful statistics automatically from publicly available YouTube videos**.

Knowing that Keenan Cornelius spends 95% of the time in open guard on the bottom allows you to study the 5% of the time that he is NOT doing that, and this system is already capable of producing valuable statistical information that may allow for large-scale study of BJJ sport strategy, and provide valuable and non-intuitive insights not available from other sources. For example, Knee-Mount is an extremely rare position and high level matches are substantially different than lower level matches in terms of positions used. While originally we trained on Black-Belt level matches, we had to change our focus to White and Blue-Belt positions in order to get enough sample images for side control and turtle positions.

Current pre-processing of the videos was simply chopping off thirty seconds from the beginning and end and then taking an image from every two seconds. However, some videos had occluded sections (judges walk in front of cameras) and sometimes more than one match was on the screen at a time, which could confuse the recognizer.

We also found that massive speed-ups of manual labeling were gained by doing the images in order, and letting whatever the previous selection was be pre-selected (since that was the most likely choice). However, occasional mistakes required laborious manual removal of an image from a bucket, and no UI was created for this as of yet. Future work would include a more powerful image management tool, which would allow you to search, move, and review labeled images.

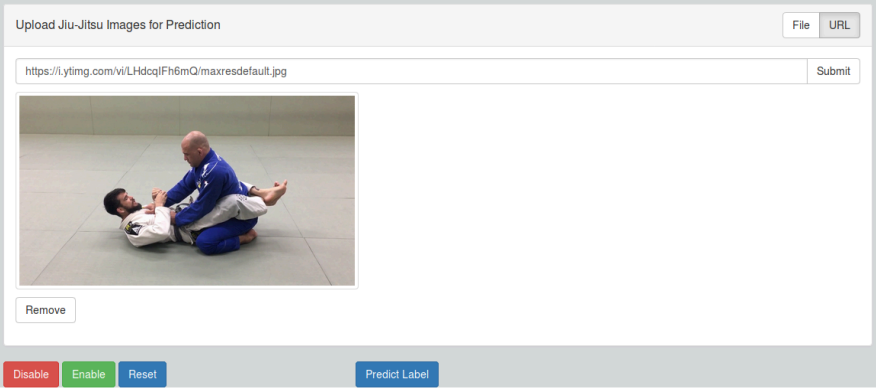
Also, many images were “invalid” which meant one of many things - and we had to specifically not train on those images (as opposed to having the machine learning train to detect invalid images as if they were a label). In some cases, the transitions between positions were labeled invalid as any one position, and in other cases, there simply was no BJJ to be found in the image. Out of our largest labeled data set of 12492 images, 5654 additional images were found to not be labelable.

Sample Results

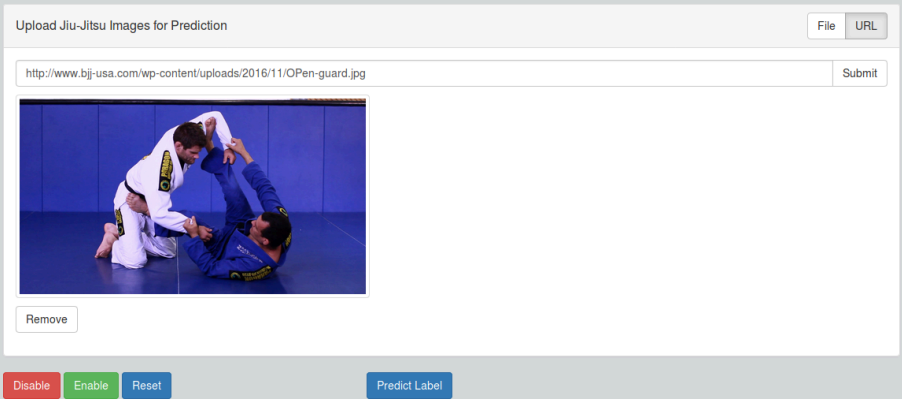
The following screenshots are of the sample website built in order to assess the model against different images quickly and easily.

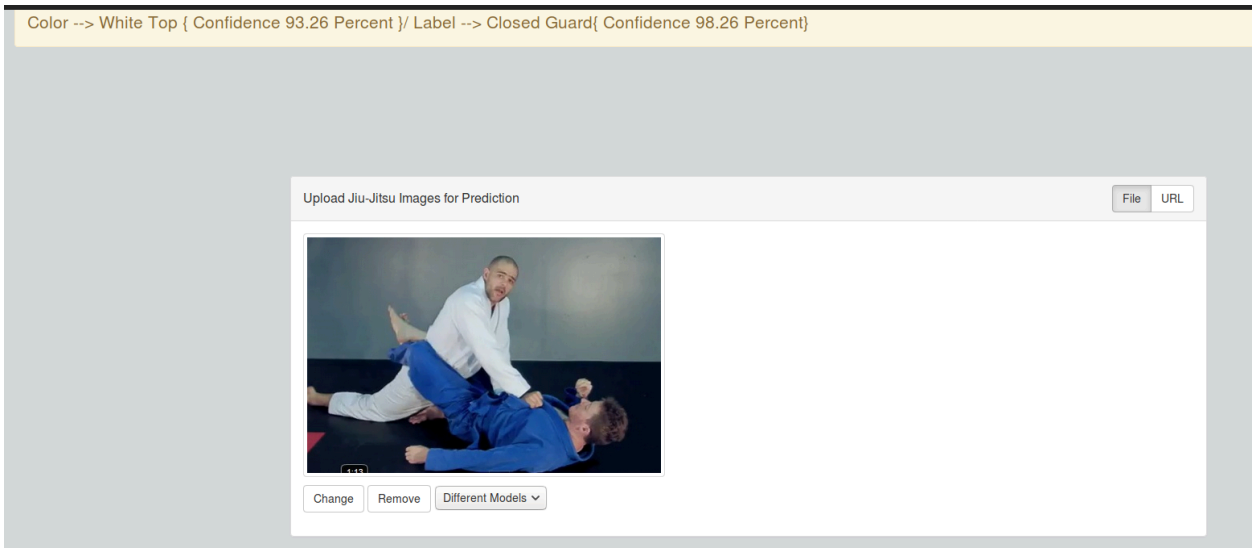
Working Live **URL** at: http://second-150623.appspot.com/

Color --> Blue Top { Confidence 80.70 Percent }/ Label --> Closed Guard{ Confidence 98.53 Percent}



Color --> White Top { Confidence 93.12 Percent }/ Label --> Open Guard{ Confidence 80.35 Percent}





Things for Future Improvements

- Balance the dataset, where all labels should be equal or at least comparable (this obviously requires more manual labeling - as many positions are quite rare!).
- Increase the number of Images in dataset and measure the increase in performance of the model.
- Once we have enough data, instead of training last few layers (fine tuning), do a deep training from scratch, this has scope where we can test different deep learning models, or create one for our use case.
- Also we can try unsupervised learning, to make clusters of different labels, and instead of labeling individual images we can label whole clusters to reduce human effort.