Synthetic Face Generation and Face Verification Models for TFHub.dev

Contact Information

Name: **Swarup Ghosh**
Email Address: **snwg@live.com**
GitHub: **swghosh** | LinkedIn: **swgghosh**

Abstract

Despite recent advances in the field of face verification and facial style transfer, implementations from scratch present a set of challenges that are likely unsuitable for many. Besides, there are limited contributions by the TensorFlow community related to such tasks.

A range of deep learning models discussed in literature which make use of various transfer learning and one-shot learning based techniques have been able to achieve human level performance on face verification tasks. Whereas, pre-trained FaceNet[14] models are already available for use, but are outside of the TensorFlow community maintained projects. For face verification task, the open source implementation of tf.keras based DeepFace[15] network can be used along with additive-Chi-squared kernel[6] to easily perform face verification on a pair of images.

Apart from the supervised classification scenario, deep generative modelling applied using style transfer approaches have been able to achieve state of the art accuracy on image generation tasks, especially synthetic faces. StyleGAN[3], a Generative Adversarial Network[7] (GAN) architecture proposed by NVIDIA Labs, has an open-source TensorFlow based implementation available.

In our proposed project, we would publish these models to TFHub.dev[20] in the form of SavedModel(s) along with the necessary documentation and Colab tutorials such that users can readily get started to apply these pre-trained models on their required scenarios. Moreover, the StyleGAN model will be ported to the TF-GAN[8] library.

Background

The computer vision research community have long been interested in the field of face recognition, pertaining to which there exists varied amount of literature related to many aspects of human faces. With the evolution and advancements of acceleration hardware (GPUs, TPUs, etc.) available for computationally expensive tasks there has been a recent rise in the field of deep learning techniques. Such techniques have been able to solve a wide array of challenging industry problems.

One of the most studied problems in face recognition literature is the task of verifying faces. Any face recognition problem with k identities (people) can be regarded as a multi-class classification problem. However, it has been found that a variety of real world problems involving face recognition can consist of a very high number of identities making it very difficult to model such problems. Thus, it has been experimentally found that such classification problems can be instead dealt with as k instances of binary classification problems. The classification task in that case would be referred to as a face verification task and would involve verifying whether a pair of images belong to the same person or not. Deep learning techniques applied to train face recognition models on large scale faces dataset are reused for face verification tasks with the help of transfer learning based approach. Researchers from around the globe have been able to achieve state of the art accuracies on face verification tasks as on the LFW[19] protocol. Such models are increasingly being used for one-shot (or few-shot) learning problems. DeepFace[1] (developed by Facebook AI Research) and FaceNet[2] (developed by Google) are two of the most widely used models that have been able to achieve high accuracy on face verification tasks also, bridging the gap to human level performance. In this project, pre-trained DeepFace model will be used that had been originally trained using softmax loss and can be easily fine-tuned on another dataset unlike triplet loss as in the case of FaceNet.

Synthetic image data generation[12] is the task of generating new images after learning the attributes and characteristics from an existing dataset. It is regarded as an unsupervised learning task which is of keen interest to many in the computer vision community. Image generation is generally used to generate stochastic variations of the information contained in the existing dataset. Data generation can either be conditional, where there is a likelihood parameter of certain input characteristic attributes based on what is expected on the generated image. Conditional Image Generation usually employs use of latent spaces to target such high level feature attributes. Alternatively, it can be unconditional where any possible random variation is observed in output images where random noise can be used as input.

According to Papers With Code[10], StyleGAN architecture proposed by NVIDIA Labs is considered as a state of the art technique for image generation using CelebA-HQ dataset based on FID quality metric. FID stands for Frechet Inception Distance and is a metric used to evaluate the quality of GAN generated images. In practice, low FID scores are indicative of a better generative model. The unsupervised separation of high level attributes that form a part of our facial structure like hair, freckles, pose, identity, etc. can be learned using generative modelling based on proposed GAN architecture as discussed in StyleGAN paper. It allows for generation of high quality images of faces that can be generated using automated methods based on latent spaces that are able to learn the factors of variations in facial structures. Apart from the generator architecture, the StyleGAN paper presents the Flickr-Faces-HQ[18] (FFHQ) dataset that consist of high quality images of human faces reflecting a wide variety in terms of pose, appearance, age, lighting, etc. and all these settings can be learned by generative models.

Although, human faces are considered to be an interesting area of study, it is notable to mention that there is a very limited set of contributions to the TensorFlow ecosystem (TensorFlow official models repo, TFHub.dev[20], TF-Addons[11], etc.) relating to computer vision applications in the facial domain. Addition of new models and examples to the TensorFlow ecosystem would attract more users and make it easy for them to build, train, fine-tune and infer such models.

Deliverables

1. New TFHub.dev SavedModel for DeepFace (pre-trained on VGG-Face2[5] dataset)
2. Addition of Additive Chi² Kernel[6] and Chi² Kernel to the TensorFlow Probability library[13]
3. Colab tutorial demonstrating Face Verification using DeepFace SavedModel
4. Colab tutorial demonstrating the fine tuning of Face Recognition task using DeepFace SavedModel
5. New TFHub.dev SavedModel for StyleGAN (pre-trained on Celeb-A-HQ or FFHQ dataset)
6. Colab tutorial demonstrating Synthetic Face Generation using StyleGAN SavedModel (with slider for adjusting various facial attributes)
7. Addition of StyleGAN to TF-GAN library[8]

Community Benefits

Newcomers in the field of machine learning take avid interest in solving problems that are related to human faces. Also, GANs have been extremely popular since its inception, and are increasingly being used to generate synthetic data for a range of tasks across various domains. Synthetic generation of proper high-quality human faces is an intuitive task given the control of synthesis on scale-specific attributes. The learning curve to get started with either face verification or applying style transfer to facial images is congenitally complex. An attempt on behalf of the larger open source community to flatten the curve would make it easier for many to get started in this field. It would make way for others to develop innovative solutions that could either employ one-shot learning or style transfer techniques meanwhile, reducing the time and efforts required to make facial systems to work.

Timelines

May 25th - May 31st (Week 0)

- Study of StyleGAN architecture
- Understand and test the existing TensorFlow implementation of StyleGAN provided by NVLabs

June 1st - June 7th (Week 1)

- Fine-tune the StyleGAN model on a suitable dataset using the existing implementation
- Refactor the Discriminator and Generator architecture of StyleGAN using tf.keras APIs

June 8th - June 14th (Week 2)

- Port the existing pre-trained weights of the model to make it suitable with tf.keras.Model
- Start with steps necessary to setup model preparation for TFHub.dev

June 15th - June 21st (Week 3)

- Write the necessary documentation for StyleGAN SavedModel
- Create a PR to publish the StyleGAN SavedModel

June 22nd - June 28th (Week 4)

- Create a Colab Tutorial that help users to get started with StyleGAN SavedModel
- Create another Tutorial that can help users to intuitively create synthetic faces by varying scale-specific settings available in the form of slider controls from within Colab

June 29th - July 5th (Week 5)

- Refine the tutorials in terms of readability and usability in order to make the tutorials engaging for users who maybe newcomers to TFHub.dev
- Publish the changes with another PR.

July 6th - July 12th (Week 6)

- Implement the Additive and Exponential Chi² kernel using tf.einsum
- Add the relevant documentation to these functions
- Contribute these kernel functions to TensorFlow Probability repository (tfp.math.psd_kernels)

July 13th - July 19th (Week 7)

- Write tests to be performed on the two kernel functions
- Create a PR to publish the kernel functions
- Port tf.keras based pre-trained DeepFace implementation to TFHub.dev

July 20th - July 26th (Week 8)

- Add the relevant documentation to allow users perform face verification task and fine tune for face recognition datasets
- Create a Colab tutorial demonstrating face verification using additive Chi² kernel as given in original DeepFace paper

July 27th - August 2nd (Week 9)

- Refine the documentation in terms of readability so as to make it easy and engaging for new users
- Publish the DeepFace SavedModel to TFHub.dev by creating a suitable PR

August 3rd - August 9th (Week 10)

- Refine the reusable aspects of the the tf.keras based Model of StyleGAN
- Convert the model into TF-GAN required format (TensorFlow Estimators API)

August 10th - August 16th (Week 11)

- Add example scripts/Colab notebooks to TF-GAN/examples.
- Create a PR to publish changes to the TF-GAN library

August 17th - August 23th (Week 12)

- Enact upon the review of the project mentor
- Modifications to existing codebase and documentation

Prior Experience

The original DeepFace implementation uses a proprietary Facebook dataset and there were no other openly available implementations. I had been able to replicate the model training process[15] mentioned on paper, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification" by Taigman et al. using openly available VGGFace2 dataset and by use of Cloud TPU resources provided by TensorFlow Research Cloud (TFRC) program. Over the years, I have found avid interest in open source while replicating proprietary benchmarks into the open source space. Have also applied deep generative modelling techniques to a few areas as a part of exploring computer vision in general. My background includes work on applying machine learning techniques to the computer vision domain and interacting with core computational systems to create high-performance cloud native applications. Previously, I have made a few contributions to the TensorFlow Addons repository and made bug reports and fixes to the main TensorFlow repository. My interests include programming across various paradigms with the aim to develop frameworks with a concrete API design. It would be a pleasure to make more contributions to the TensorFlow ecosystem even outside of the GSoC tenure.

References

[1] Taigman, Yaniv, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. "Deepface: Closing the gap to human-level performance in face verification." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1701-1708. 2014.

[2] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 815-823. 2015.

[3] Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4401-4410. 2019.

[4] Karras, Tero, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and improving the image quality of stylegan." arXiv preprint arXiv:1912.04958 (2019).

[5] Cao, Qiong, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. "Vggface2: A dataset for recognising faces across pose and age." In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 67-74. IEEE, 2018.

[6] Li, Ping, Gennady Samorodnitsk, and John Hopcroft. "Sign cauchy projections and chi-square kernel." In Advances in Neural Information Processing Systems, pp. 2571-2579. 2013.

[7] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in neural information processing systems, pp. 2672-2680. 2014.

[8] tensorflow/gan: Tooling for GANs in TensorFlow

[9] TensorFlow Module: tf.keras

[10] Papers With Code, State-of-the-Art in Machine Learning

[11] tensorflow/addons: Useful extra functionality for TensorFlow 2.x maintained by SIG-addons

[12] Papers With Code, SOTA: Image Generation

[13] tensorflow/probability: Probabilistic reasoning and statistical analysis in TensorFlow

[14] davidsandberg/facenet: Face recognition using Tensorflow

[15] swghosh/DeepFace: Keras implementation of the renowned publication "DeepFace: Closing the Gap to Human-Level Performance in Face Verification" by Taigman et al. Pre-trained weights on VGGFace2 dataset.

[16] NVlabs/stylegan: StyleGAN - Official TensorFlow Implementation

[17] NVlabs/stylegan2: StyleGAN2 - Official TensorFlow Implementation

[18] NVlabs/ffhq-dataset: Flickr-Faces-HQ Dataset (FFHQ)

[19] Huang, Gary B., Marwan Mattar, Tamara Berg, and Eric Learned-Miller. "Labeled faces in the wild: A database for studying face recognition in unconstrained environments." 2008.

[20] tensorflow/hub: A library for transfer learning by reusing parts of TensorFlow models.