CS 275 Project Report - "Script to 3D Model"

Sonia Jaiswal
UID: 805525305
Department of Computer Science
University of California at Los Angeles
soniajaiswal@g.ucla.edu

Diplav
UID: 605627748
Department of Electrical and Computer Engineering
University of California at Los Angeles
diplav@g.ucla.edu

Srinath Naik
UID: 605525467
Department of Computer Science
University of California at Los Angeles
srinath@cs.ucla.edu

Tanmay Sanjay Hukkeri
UID: 205525129
Department of Computer Science
University of California at Los Angeles
tanmayhukkeri@ucla.edu

Arunachalam Chidambaram
UID: 505430144
Department of Electrical and Computer Engineering
University of California at Los Angeles
arun808@ucla.edu

1 Abstract

Artificial synthesis of human facial models continues to be an active area of research, sitting at the intersection of Computer Vision, Computer Graphics and Artificial Life. Spanning across various fields from medicine to film, the key task remains generating expressive and accurate three-dimensional renderings of human faces. Interestingly, facial synthesis has also seen active interest in the two-dimensional domain, with an increasing use of Generative Adversarial Network (GANs) to generate and manipulate human face renderings. This project thus attempts to take a step towards combining the research in these various disciplines, integrating the fields of Natural Language Processing, 2-D text guided manipulation, and 3-D model capture and rendering. Specifically, this project attempts to create a "text-to-3D-model" pipeline, generating three-dimensional faces to convey the emotion carried by a piece of text.

2 Introduction

The problem of rendering accurate, expressive and realistic human faces in a three-dimensional space is non-trivial. Recent years have seen several attempts towards three-dimensional face generation, such as DECA[1], EMOCA[2] and 3DFaceGAN[3]. The key difficulty in all these attempts remains accurately capturing details such as texture and expression. These 3-D generated renderings see wide use in a large number of fields, from aiding in medical diagnosis, to producing artificial characters and actors in a movie. The latter is particularly interesting, with a far reaching goal of AI generated three-dimensional representations that can show the emotional range of a real actor. Development teams , such as those at Pixar currently use three-dimensional models of humans in several of their scripts and films , such as the Toy Story series and the 2015 movie Inside Out. However, correlating the actions and expressions of these models with the directive specified in the script involves a large amount of human-effort. A far-reaching goal could thus envision the use of AI to auto-generate 3D

Figure 1: Human 3D models in the Pixar movie Inside Out[4]

context-sensitive models, reacting and acting based on the script without any human intervention.

An additional effort associated with 3D rendering involves manually drawing up the various models and representations for conveying various emotions. Each model requires hours of effort to capture an image conveying the required emotion. This opens the door for the use of Generative Adversarial Networks. In particular, we propose that the use of text-guided image manipulation could help support this task, as it requires only one base image that can be manipulated into pictures of various emotions simply with the help of a guiding text. These images can then be used to generate the corresponding 3D renderings using tools such as DECA[1]. This project attempts to take an initial step in this direction, treating individual tweets and pieces of text as our script. The goals of this project are thus outlined as below:

- To parse a text input and retrieve from it the emotion conveyed, with the help of a pre-trained and fine-tuned model.(Emotion Detection)
- To use a base image of a human face and manipulate it to convey the emotion previously detected (2D text-guided image manipulation)
- To utilise the DECA model to convert this image to a 3D rendering and thus obtain a 3D model conveying the captured emotion.

3 Literature Survey

The problem of generating expressive and rich three-dimensional models has garnered a lot of attention over the years. Starting from the pioneering work of Parke in 1974 in 3D reconstruction, this domain has seen active research. As have equally the fields of natural language processing on sentences and image generation and manipulating using Generative Adversarial Networks. In this section, we provide some context on the work done so far in these respective domains.

3.1 Text Analysis and Emotion Detection

Natural Language Processing tasks have taken center-stage in the AI domain, being one of its core tenets along with Computer Vision and Speech Processing. The inception of models like the Recurrent Neural Network[5] and Transformers[6] has seen a surge in the development of tools for tasks like text-analysis, sentiment analysis and language modelling. In particular, text-analysis has seen active research. Sentiment analysis, remains one of the standards as an introductory task in the NLP domain, with key datasets such as SST-2[7] and IMDb[8], and models such as BERT[9] and RoBERTa-base[10]. An extension of this is emotion detection, where-in instead of predicting whether a given sentence is negative or positive, the sentence is classified according to the emotion conveyed, such as joy or anger. Several attempts have been made towards successful emotion classification, such as NRC-Lexicon (EMOLEX)[11][12] and RoBERTa-Base[10]. In this project however, we choose to work with a unified model, namely the T5-model[13], on account of its success and its scalablilty to other tasks.

3.2 Image Synthesis and Manipulation

An exciting new area of research is the use of Generative Advesarial Networks, or GANs to perform tasks such as Image Generation and Image Manipulation. Recent years have seen several works in

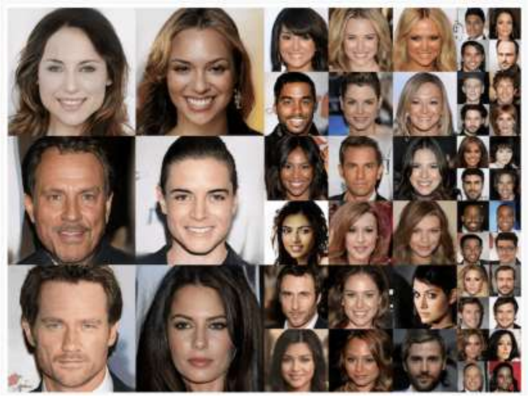


Figure 2: Human faces generated by PGGAN

this domain, with models such as DCGAN[14] and BigGAN[15] demonstrating successful image generation capability. A model of interest in this respect is PGGAN[16], being one of the first to generate images of human faces, as shown in Figure 2. These models however, all focus on generating random images, with significant variation on each run.

Instead, we shift our focus towards manipulating a base image, in order to obtain a new image which maintains a majority of features of the base image. Models such as StackGAN[17] and more recent works such as FacialGAN[18], ControlGAN[19] and ManiGAN[20], all aim to manipulate images using provided information. While [18] relies on a target image and label to perform manipulation, [19] and [20] align more with the task at hand, using text to attempt to manipulate images. However, both [19] and [20] were predominantly developed for modifications on the Caltech-UCSD Birds-200-2011[21] and COCO[22] dataset, and do not generalise well to the use case of human faces. In the next section, we showcase the limitations of these models on the task at hand, and also describe TediGAN[23][24], a recent model trained for the purpose of manipulating human faces.

3.3 3D Shape Generation and Detail Reconstruction

In recent years, there have been tremendous advances in 3D shape generation. While some of this work includes traditional 3D representations such as point clouds[25], voxels[26] and meshes[27], several approaches have been proposed to use implicit surface and volume representations for high-quality and scalable representations. However, most of this work focuses on generating rigid objects, a relatively simple task compared to the generation of articulate, morphable shapes such as the human face and body.

Another body of work aims to reconstruct faces with “mid-frequency" details. Common optimization-based methods fit a statistical face model to images to obtain a coarse shape estimate, followed by a shape from shading (SfS) method to reconstruct facial details from monocular images or videos[28]. These approaches are slow, the results lack robustness to occlusions, and the coarse model fitting step requires facial landmarks, making them error-prone for large viewing angles and occlusions.

Unlike prior work in this domain, our project seeks to eliminate the need for an intermediate image seed, and rather seeks to create an end-to-end pipeline from text to 3D Model, via the use of image synthesis as an intermediary.

4 Methodology

In this section , we provide a detailed description of the three-step pipeline described in Section 2. For each section, we describe the models surveyed, the final model chosen along with supporting rationale.

4.1 PART 1: Extracting emotion from text

The first step in the process involved extracting the emotion from a given piece of text. Given the abundance of models that perform emotion classification, we opt to utilise an existing pre-trained model. To achieve this we experimented with various existing models, selected the one most relevant to our task and further fine-tuned the same. The below sections provide detail on the same.

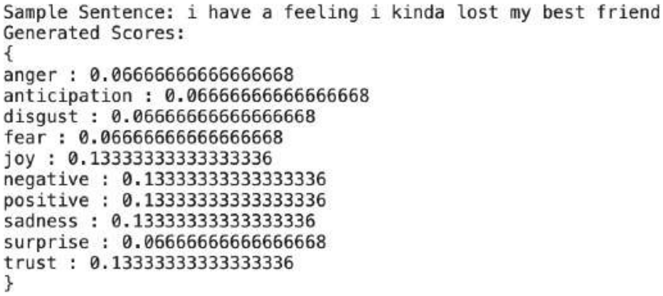
4.1.1 Twitter-roberta-base-emotion

We first experimented with Twitter-roberta-base-emotion[10], which is a RoBERTa-base model trained on 58M tweets and fine tuned for emotion recognition with the TweetEval benchmark[29][30]. RoBERTa is a transformers model pre-trained on a large corpus of English data in a self-supervised fashion. This means it was pretrained on the raw texts only, with an automatic process to generate inputs and labels from those texts. More precisely, it was pre-trained with the Masked language modeling (MLM) objective. Taking a sentence, the model randomly masks 15% of the words in the input then run the entire masked sentence through the model and has to predict the masked words. This is different from traditional recurrent neural networks (RNNs) that usually see the words one after the other, or from autoregressive models like GPT[31] which internally mask the future tokens. It allows the model to learn a bidirectional representation of the sentence.

A key limitation of this model was that it could only provide 4 emotions, defined as joy, optimism, anger and sadness by the TweetEval benchmark. Given our goal to generate varied images with different emotions, as well for ease of scalabilty, we decided to opt for a more expressive dataset, namely the Dair-AI dataset[32], which provided 6 emotions : sadness, joy, love, anger, fear and surprise . This model would thus require to be trained from scratch on the new dataset, which would be computationally expensive.

4.1.2 NRC Word-Emotion Association Lexicon

The NRC Emotion Lexicon [11][12] is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The annotations were manually done via crowdsourcing. This approach is pretty simple and easy to understand and use. We experimented by writing the code to get the emotions out of sample text using NRCLex and observed the results for some of the sentences. Figure 3 below demonstrates on the output results with the NRC benchmark called.



```
Sample Sentence: i have a feeling i kinda lost my best friend
Generated Scores:
{
anger : 0.06666666666666668
anticipation : 0.06666666666666668
disgust : 0.06666666666666668
fear : 0.06666666666666668
joy : 0.13333333333333336
negative : 0.13333333333333336
positive : 0.13333333333333336
sadness : 0.13333333333333336
surprise : 0.06666666666666668
trust : 0.13333333333333336
}
```

Figure 3: Sample result generated by EMOLEX

From the above example one can see the results are confusing. It classifies the text equally as trust, positive and negative also. Since the dictionary is based on the basis of words , it doesn't take consideration of semantic relations and is thus not useful for our use case. Additionally, since EMOLEX operates off a dictionary based search, its runtime far exceeds the prediction runtime of a neural network, making it less ideal for our use-case.

4.1.3 Text-to-Text Transfer Transformer

So the final approach that we used in our model is based on T5[33]. T5 stands for Text to Text Transfer Transformer. The model is based on the notion of Transfer Learning, in which the entire model is pre-trained on a data-rich task, which ideally helps the model to develop general purpose knowledge that can be utilized by downstream tasks. Modern techniques for transfer learning in NLP use unsupervised learning on easily available, abundant unlabeled data to pre-train for better performance and scalability. In this approach, every text processing problem is treated so that the model inputs and outputs are both text. This allows to apply the same model, objective, training procedure, and decoding process to every task it is applied to.

A key motivation for using the T5-base finetuned model lie in its scalablity. As the model is unified and suited for most natural language tasks, it provides an easy upgrade path from generation of 3D models based on emotion, to other tasks such as generation based on descriptions.

4.1.4 Architecture and Experimental Setup

As described in the previous section, we thus use a pre-trained t5-base architecture and fine-tune it on the Dair-AI dataset[32], following the approach listed in [34] . The model is based off of a standard encoder-decoder architecture as shown in Figure 4.

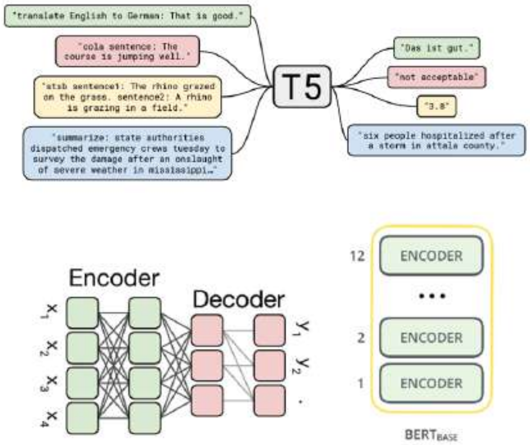

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi..."
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."
Encoder
Decoder
x_1
x_2
x_3
x_4
y_1
y_2
12
ENCODER
...
2
ENCODER
1
ENCODER
BERT$_{BASE}$

Figure 4: T5 architecture as described in [13]. Top: Overview of model functionality. Bottom left: Encoder-Decoder architecture[13]. Bottom right: single stack in the encoder/decoder, mirroring BERT base[35]

This model uses a standard encoder-decoder Transformer as proposed by [36]. The design involves both encoder and decoder models that are similar in size and configuration to a "BERTBASE" stack. 12 blocks comprise the encoder and decoder with each block containing self-attention, optional encoder-decoder attention, and a feed forward network. Inside each block, the mentioned feed forward network consists of a dense layer with an output dimensionality of dff=3072 and is followed by a ReLU non-linearity and another dense layer. For all the attention mechanisms, the key and the value matrices have an inner dimensionality of dkv=64 and all attention mechanisms have 12 heads. The other sub-layers and embeddings within have a dimensionality of dmodel=768. In totality, this amounts to 220 million parameters, which is roughly twice as big as the parameter of "BERTBASE" since the baseline model contains two layer stacks instead of one. For regularization, a dropout probability of 0.1 is used everywhere applicable.

The work in [34] demonstrates the application of the T5 model to the task of emotion detection, and we extend on this, attempting to finetune further for a total of 5 epochs. We note however, that between 3-5 epochs, there was not much of a shift in accuracy. It caps of at an accuracy of 93%

4.2 PART 2: Emotion to 2D Image

The second step involved the generation of a 2D human face image, that carried the emotion detected in the previous step. The input image is supplied to the model, thought to be the "user" writing the piece of text or tweet. As in the previous section, we again survey a few existing models that perform the required text-guided image manipulation, and optimise the chosen model based off qualitative analysis.

4.2.1 FacialGAN

We first explore the utility of FacialGAN[18], a framework based off of rich style transfer and a user interface based interactive manipulation approach. The model makes use of a target image and a guiding segmentation mask image , and provides an interactive UI to modify features such as nose, eyes and mouth, and transfer information from the above into the original image. However, our task requires that most of the structural information of the input image be preserved, and only the emotion conveyed be changed. Figure 5 demonstrates an attempt at trying to achieve this.

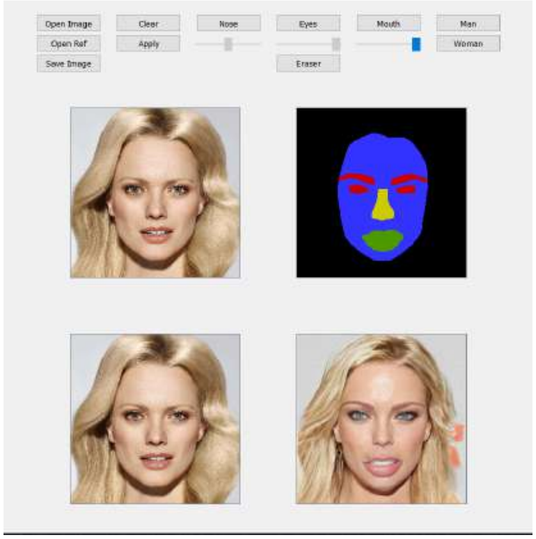


Figure 5: Attempt as using FacialGAN for emotion manipulation

As shown in Figure 5, we attempt to keep both the input image and the reference image the same, in order to try and preserve information. We then attempt to maximise the size of the eyes and the mouth, to try and convey the emotion "surprised" . As shown in the figure, the output is quite lacklusture, and thus we conclude that the above model is not well-suited for our task.

4.2.2 ManiGAN

Our next attempt involved working with ManiGAN[20], a recent work that performs text-guided image manipulation. The model derives from ControlGAN[19], and is predominantly trained on the Caltech-UCSD Birds-200-2011[21] and COCO[22] dataset. The model makes use of an image-encoder and a text-encoder to utilise information from both inputs in order to try and generate an output. Figure 6 demonstrates an attempt to use the COCO-trained ManiGAN for emotion manipulation.

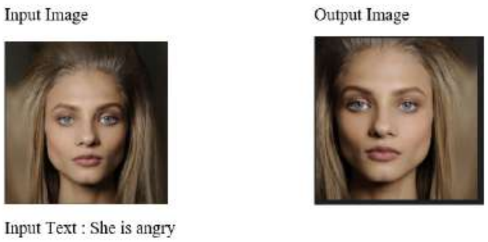


Figure 6: Attempt at using MANIGAN for emotion manipulation

As we observe , the outputs from the COCO-pretrained MANIGAN do not capture emotion information well. We instead attempt to finetune this model on the CelebA-HQ [35] dataset, however, the dataset is missing key embedding information that prevents the finetuning from happening. So we depart from attempting to use ManiGAN, and instead choose the next model, namely TediGAN.

4.2.3 TediGAN

The final model we survey, and the one this proposed project uses is the TediGAN[23][24] model. This model synthesizes a photo-realistic 2D facial image based on textual description, and is based on using pretrained GAN models. It makes use of the process of StyleGAN inversion , along with a text encoder in order to obtain the required latent codes in the latent space. The model is trained on the Multi-Modal CelebA-HQ dataset[37], which contains image as well as textual inputs in order to faciliate the training of the model. Figure 7 showcases the architecture as proposed in [23].

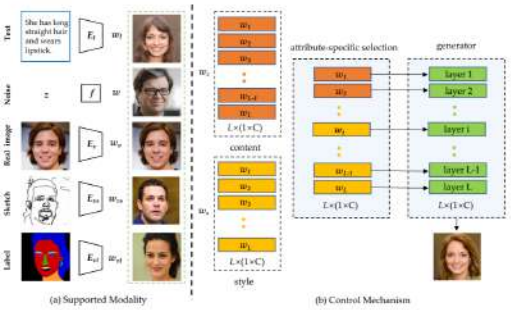


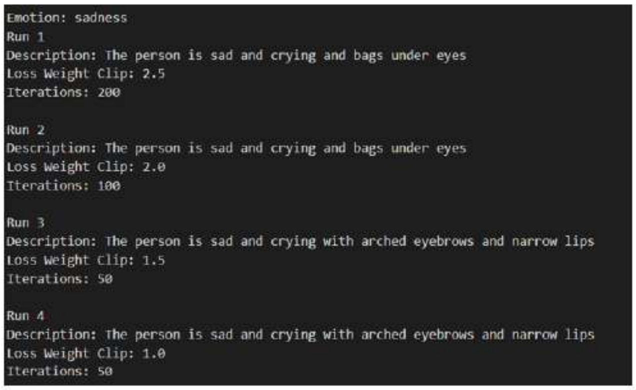
Figure 7: TediGAN framework proposed by [23]

The key components of this architecture are the Image generators and encoders, both based of the in-domain GAN inversion technique proposed by [38]. This method involves treated the trained encoder as a regularizer, to allow the latent codes predicted to land better inside the required latent space. Equally important is the text-encoder, which uses a visual-learning linguistic similarity module based approach to map both the image and the text onto a common embedding space.

4.2.4 Experimental Setup

As discussed in the previous section, we thus make use to the pretrained TediGAN model, making use of the available pre-trained weights for the image generators and encoders, as well as for the text encoders. For the purpose of this experiment, we intially attempted to fine-tune the image generator on datasets such as FFHQ[39]. Experimental analysis revealed limited support in doing so however, and training the GAN models from scratch proved infeasible on the available hardware resources. We instead thus attempt to shift our focus towards improving the output generated.

Analysing the output of the TediGAN module, we observe that the model generates well suited outputs for emotions like joy and anger, but does not do so well on emotions like surprise. This in part is due to the text labels in [23][24] which do not account for such specific emotions. We thus attempt to fine-tune the results obtained through a process of continuous application of the model. More specifically, we recursively pass the generated outputs back into the model, each time decreasing the associated loss-weight-clip and processing iterations. The idea is to perform a "repeated loss-weight-clip decay", inspired by the commonly used approach of learning-rate decay in neural networks, to help add finer detail to the generated output. We also tweak the descriptions at each stage, adding more and more descriptors on specific facial parts to construct the emotion piece-by-piece, and thus overcome the limitation of a lack of descriptors in the dataset. In total, we pass the image through the model 4 times. In Section 5.2, we demonstrate some of the results of using this approach, and showcase the improvement of the "repeated loss-weight-clip decay" approach over a single-pass use of the model. Figure 8 below describes one such example.



```
Emotion: sadness
Run 1
Description: The person is sad and crying and bags under eyes
Loss Weight Clip: 2.5
Iterations: 200

Run 2
Description: The person is sad and crying and bags under eyes
Loss Weight Clip: 2.0
Iterations: 100

Run 3
Description: The person is sad and crying with arched eyebrows and narrow lips
Loss Weight Clip: 1.5
Iterations: 50

Run 4
Description: The person is sad and crying with arched eyebrows and narrow lips
Loss Weight Clip: 1.0
Iterations: 50
```

Figure 8: Figure demonstrating the "repeated loss-weight-clip decay". Top: The parameters involved in generating input from output. Bottom Left: Original Image. Bottom Right: Generated Image for Emotion: Sadness

4.2.5 Image Alignment

We also make a small mention about image alignment. The TediGAN framework is trained on datasets such as Celeb-A-HQ and FFHQ, all of which contain images aligned according to face landmarks. Thus, in order to utilise any web-scraped, out of dataset image, the image would first need to be aligned. We make use of a separate approach provided by [40] to do the same. Figure 9 below demonstrates using the above described TediGAN procedure on an out of dataset image.

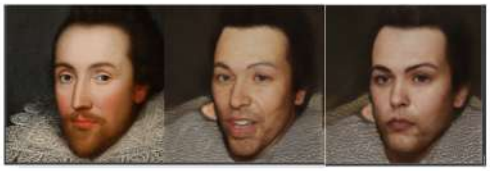


Figure 9: Figure demonstrating the output of "repeated loss-weight-clip decay" based TediGAN to an image of Shakespeare.

4.3 PART 3: 2D Image to 3D Model

The final step in the pipeline involves creating and rendering a 3D model from the previously generated 2D images. We attempt to use two different approaches towards the same.

4.3.1 DECA

In order to convert the 2D emotional image generated in the previous step to a realistic 3D animatable model, we started with DECA(Detailed Expression Capture and Animation)[], which is trained in the wild in an unsupervised way to realistically capture the nuances of the coarse 3D structure as well as wrinkle details. We leverage this work to generate 3D models corresponding to the emotionally modified(generated) input image.

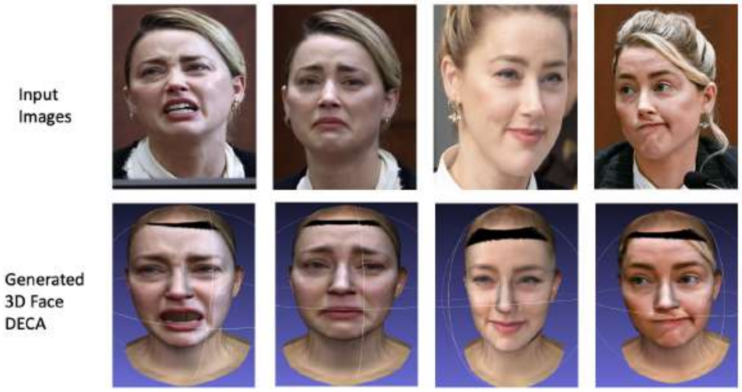


Figure 10: Top row: Input image with challenging expressions. Bottom row: 3D face reconstruction result using DECA model.

DECA uses an encoder-decoder kind of structure, where the encoder projects the input image into two distinct latent space variables corresponding to structure and detail respectively. The structure latent variables are used to create a coarse 3D structure whereas the detail variables are used to generate the wrinkles around them which capture the inherent expression. Later, both these structures are joined together to form the 3D model. This final model is projected into 2D back and reconstruction loss is used to train the model.

We have tested the state of the art DECA model on some of our emotionally manipulated images and the results are reasonably good. In order to make it even more effective and realistic, additionally we have setup fine-tuning the DECA model on FFHQ-Dataset as much of the images we are using in previous module are from FFHQ.

FFHQ has around 70000 images of Flickr Faces. A single epoch of training on this amount of data is taking a long time (approximately > 2 hours for 1 epoch) with the computational resources we have(ColabPro). Hence, we have decided to fine tune on a sampled set from FFHQ. We have randomly selected 10,000 images from the original Dataset to create FFHQ-10K dataset. DECA internally uses face alignment module to get the 68 key landmarks corresponding to a face and use them in training, so we had to do a similar pre-processing on the 10K Dataset to generate the landmarks and save them additionally. We have used the above created dataset and fine-tuned DECA on top of it. Some of the results can be seen in Figure 10.

4.3.2 EMOCA

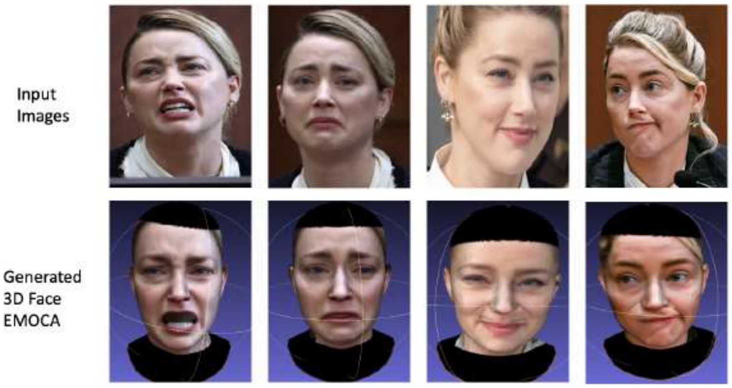


Figure 11: Top row: Input image with challenging expressions. Bottom row: 3D face reconstruction result using EMOCA model.

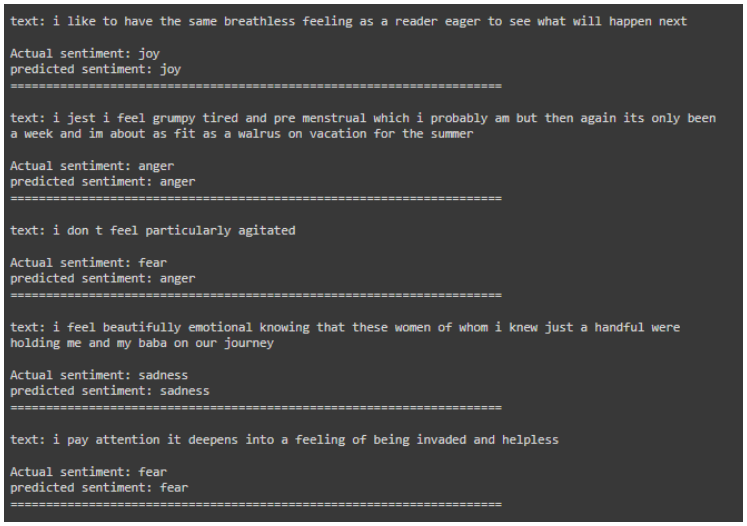
In addition to DECA we also tried EMOCA model to generate the 3D face model from monocular image which is an extension of the DECA model and uses a novel deep perceptual emotion consistency loss during training, which helps ensure that the reconstructed 3D expression matches the expression depicted in the input image.The DECA architecture is augmented with an additional trainable prediction branch for facial expression and is trained in a self-supervised fashion on an emotion rich image dataset[41]. Main goal of our experiment was to finetune EMOCA so that emotion in the input is expressed more clearly in the reconstructed 3D face. So to acheive this we further fine tuned EMOCA on the AffectNet dataset [41] which contained in-the-wild emotional face images and evaluate emotion recognition accuracy based on the 3D reconstruction.

5 Experimental Results

In this section , we provide a component-wise result analysis, with qualitative results demonstrating the outputs and quantitative results helping substantiate claims. In our video, we demo a run of the complete pipeline, as well as several 3D rendered outputs.

5.1 Emotion detection: T5-base Results

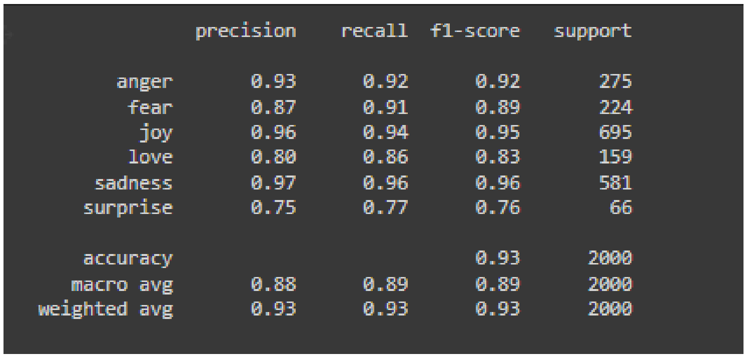
As described in Section 4.1, we utilise the T5 base architecture, and fine-tune it for 5 epochs on the DAIR-AI dataset. We note a marginal improvement in accuracy over [34], with a jump from 92.2% to 92.65%, and no real improvement between epochs 3 to 5. This seems to indicate a plateau on the effectiveness of fine-tuning, and demonstrates the need for other means to improve the accuracy, such as collecting more data. Figure 12 below demonstrates the output of the t5 module on various test examples. As we can see, the model does well on most test examples, but occasionally falters, particularly when the texts convey emotions such as fear and surprise, which are harder to detect.



```
text: i like to have the same breathless feeling as a reader eager to see what will happen next

Actual sentiment: joy
predicted sentiment: joy
====================================================================

text: i jest i feel grumpy tired and pre menstrual which i probably am but then again its only been
a week and im about as fit as a walrus on vacation for the summer

Actual sentiment: anger
predicted sentiment: anger
====================================================================

text: i don t feel particularly agitated

Actual sentiment: fear
predicted sentiment: anger
====================================================================

text: i feel beautifully emotional knowing that these women of whom i knew just a handful were
holding me and my baba on our journey

Actual sentiment: sadness
predicted sentiment: sadness
====================================================================

text: i pay attention it deepens into a feeling of being invaded and helpless

Actual sentiment: fear
predicted sentiment: fear
====================================================================
```

Figure 12: Output of t5 emotion detection on out of dataset examples

This is further cemented by the metric report shown in Figure 13 below, which demonstrates lower precision for the classes surprise and love.



```
              precision    recall  f1-score   support

       anger       0.93      0.92      0.92       275
        fear       0.87      0.91      0.89       224
         joy       0.96      0.94      0.95       695
        love       0.80      0.86      0.83       159
     sadness       0.97      0.96      0.96       581
    surprise       0.75      0.77      0.76        66

    accuracy                           0.93      2000
   macro avg       0.88      0.89      0.89      2000
weighted avg       0.93      0.93      0.93      2000
```

Figure 13: Metric Report

5.2 2D Image Generation: Results

Figure 14 below demonstrates the comparison between images generated using a single-pass through the TediGAN framework, and images generated using the "repeated loss-clip-weight" approach.

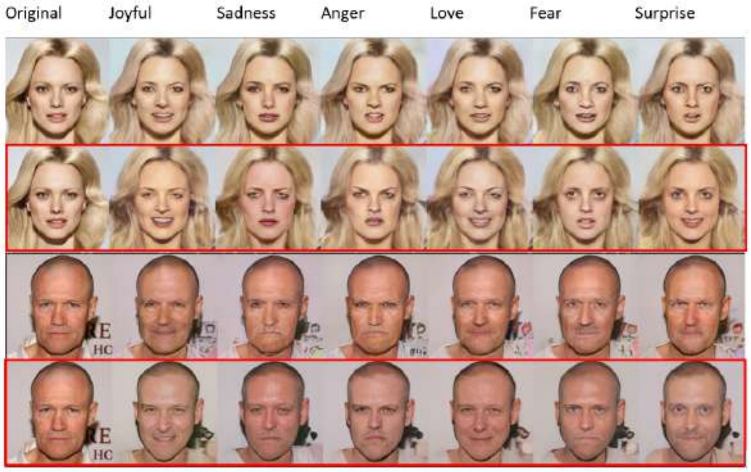
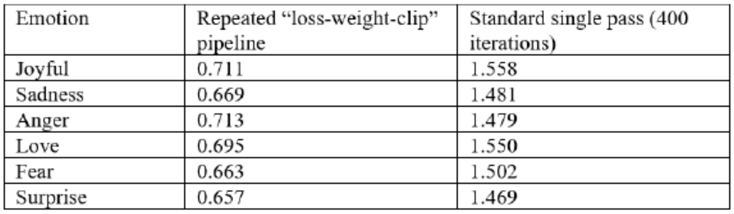


Figure 14: Results of single-pass TediGAN vs "repeated loss-clip-weight" TediGAN. "Repeated loss-clip-weight" rows are highlighted in red.

As we can see from the above figure, in most cases, the "repeated loss-clip-weight" approach demonstrates more expressive features, able to capture the emotion more clearly. This makes it easier for 3D rendering models like DECA and EMOCA to render 3D models with significant differences. It is also worth noting that the extra iterations and multiple passes result in some shift in texture from the original image, however, we believe that this can be corrected for via stronger image generation models and a more robust search for the optimal parameters in this repeated approach.

Figure 15 below also demonstrates a quantitative analysis using the overall loss after generation as a metric. We observe that for all emotions, the "repeated loss-clip-weight" approach shows much lesser loss than the standard single pass approach, even when keeping the number of epochs equivalent.



Emotion	Repeated “loss-weight-clip” pipeline	Standard single pass (400 iterations)
Joyful	0.711	1.558
Sadness	0.669	1.481
Anger	0.713	1.479
Love	0.695	1.550
Fear	0.663	1.502
Surprise	0.657	1.469

Figure 15: Loss Analysis for both techniques

5.3 3D Model Generation: EMOCA vs DECA

EMOCA is better at capturing emotional expression when compared to DECA. The EMOCA[2] reports an improvement of 0.05 in a test conducted on 3D reconstruction using both EMOCA and DECA. We can also observe from Figure 10, 11 that EMOCA comparatively better at the emotions.

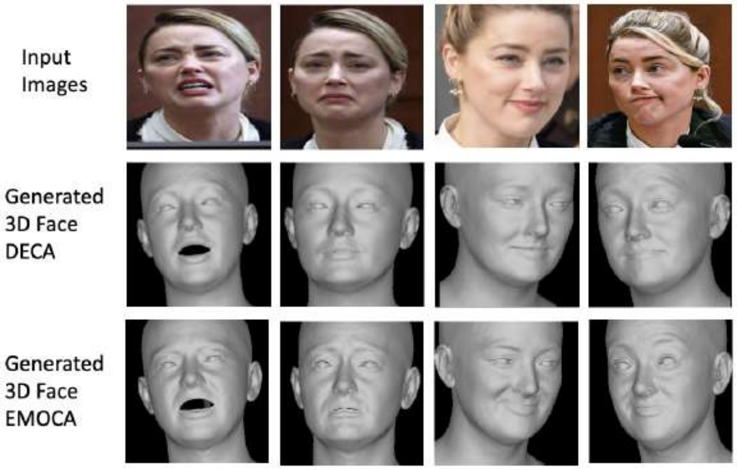


Figure 16: 3D reconstruction comparison for DECA and EMOCA

We provide a visual comparison of the coarse shape reconstruction of DECA and EMOCA in Figure 16. We can observe that EMOCA outperforms DECA on 3D coarse reconstruction and expression are captured more vividly in EMOCA as compared to DECA. We can observe EMOCA is able to capture the fine details such as face lining, wrinkles which were missed by DECA.

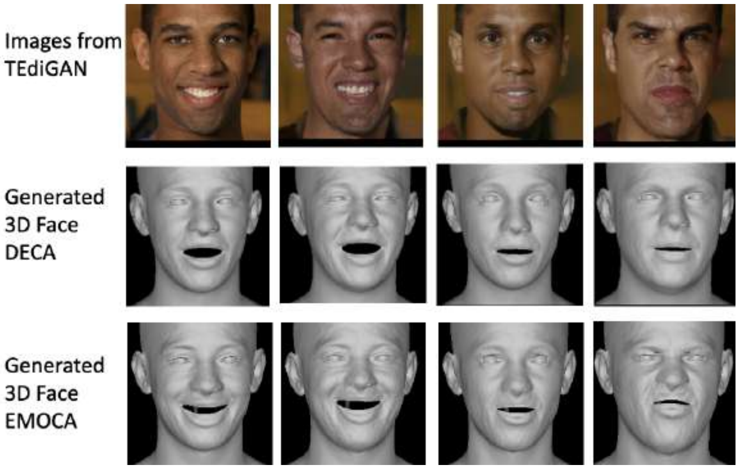


Figure 17: 3D reconstruction comparison for DECA and EMOCA on Tedigan generated images.

We also visualized the comparative results on TediGan generated image on various emotion to check how much both the model are able to generalize on unseen data distribution and able to capture the emotional content of the original image. The result is shown in Figure 17. Both the model are able to generalize well on unseen dataset. However EMOCA is even able to capture very fine expression details such as facial hair details of the original image in the reconstructed expression.

6 Conclusion and Future Work

The proposed project was able to successfully demonstrate a pipeline that went from text inputs to corresponding emotion-aware 3D models. A defining aspect of the project involved the use of GAN based models such as TediGAN to eliminate the need to generate individual images for each emotion. Through the project, we also conduct a rigorous survey of the various tools and models available for each stage of the pipeline, and present our case for our choices in models. Future directions for the project could include tasks such as better 2D image generation using more powerful GANs, and conversion towards other such text-to-model tools such as Description-to-3D-Model. The report, video and code files are available here

References

[1] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics (ToG), Proc. SIGGRAPH*, 40(4):88:1–88:13, August 2021.

[2] Radek Daněček, Michael J Black, and Timo Bolkart. Emoca: Emotion driven monocular face capture and animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20311–20322, 2022.

[3] Stylianos Moschoglou, Stylianos Ploumpis, Mihalis Nicolaou, and Stefanos Zafeiriou. 3dfacegan: Adversarial nets for 3d face representation, generation, and translation. 05 2019.

[4] Pluralsight. Your first look at character design in pixar's new "inside out", Jun 2017.

[5] Zachary Lipton. A critical review of recurrent neural networks for sequence learning. 05 2015.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[7] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[8] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[11] Saif M. Mohammad and Peter D. Turney. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[12] Saif Mohammad and Peter Turney. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Los Angeles, CA, June 2010. Association for Computational Linguistics.

[13] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[14] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[15] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[17] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5908–5916, 2017.

[18] Ricard Durall, Jireh Jam, Dominik Strassel, Moi Hoon Yap, and Janis Keuper. Facialgan: Style transfer and attribute manipulation on synthetic faces, 2021.

[19] Minhyeok Lee and Junhee Seok. Controllable generative adversarial network. *IEEE Access*, 7:28158–28169, 2019.

[20] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7880–7889, 2020.

[21] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.

[23] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[24] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Towards open-world text-guided face image generation and manipulation. *arxiv preprint arxiv: 2104.08910*, 2021.

[25] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. 2017.

[26] Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction, 2016.

[27] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation, 2018.

[28] Luo Jiang, Juyong Zhang, Bailin Deng, Hao Li, and Ligang Liu. 3d face reconstruction with geometry details from a single image. *IEEE Transactions on Image Processing*, 27(10):4756–4770, 2018.

[29] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online, November 2020. Association for Computational Linguistics.

[30] Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17, 2018.

[31] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[32] Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.

[34] patil suraj. exploring-t5, May 2020.

[35] Bert base vs bert large.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[37] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[38] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[39] Tero Karras, Samuli Laine, and Timo Aila. Flickr faces hq (ffhq) 70k from stylegan. *CoRR*, 2018.

[40] Peter Baylies. stylegan-encoder, Sep 2020.

[41] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2017.