Text to Motion Database

Test Report

Brendan Duke
Andrew Kohnen
Udip Patel
David Pitkanen
Jordan Viveiros

March 27, 2017

Contents

1 Preface

1.1 Revision History

Table 1.1: **R**evision History

Date	**V**ersion	**N**otes
March 14, 2017	0.0	File created
March 23, 2017	0.1	Initial template completed
March 26, 2017	0.2	Completed tables and organized sections

1.2 List of Figures

This document does not utilize figures in order to display our results.

List of Tables

2 Introduction

2.1 Purpose of Document

The purpose of this manuscript is to document the testing that has been performed on the Text-to-Motion web application. The testing that has been performed largely follows the test plan that was provided in the Test Plan document.

2.2 Scope of Testing

Following good design procedures, the Text-To-Motion web application has been modularized into 3 conceptual modules. There is a web framework that consists of a front end and allows users to access the Text-To-Motion features and to input new data. There is also a back-end, which consists of two separate components: a database module and a deep-learning module.

Following the application's modular decomposition, the tests have been decomposed into 3 conceptual tests.

The first stage of testing, the Proof of Concept testing, was done to ensure that a minimal set of milestones were reached. These milestones were set to define a minimal viable product, namely the proof of concept. Here we test to see that all of the software modules mentioned above have been created and these modules can work together. So, this range of tests involves system and unit testing.

The second set of tests is Solution Constraints testing. In this set of tests, the performance of the deep learning algorithm is rigorously tested and quantified, and different formats of data that can be inputted are also tested. We also test how easy the project is to build from in our recommended environment.

In the final set of tests, we test the functional and non-functional requirements of the finished product. Examples of these requirements include: usability, supported video encodings, and maintainability of the web site.

The final section in this document includes a traceability table that helps to organize and explain how tests are connected to the requirements that were created in the software requirements document we created earlier.

3 Preliminary Testing

3.1 Proof of Concept

3.1.1 Functional website for pose estimation

Description

This is a baseline test to make sure the website is running, appears in a web browser when its URL is entered, and that pictures can be uploaded and viewed on the site.

Put the proper URL into the browser and click on each of the links to make sure that no error occurs, meaning that there are no excessive wait times and that the browser does not crash. Make sure there are human poses shown on the site.

Results

All of the tests passed. The website was accessible through the browser, and all hyperlinks were functional. Poses were displayed on the ImagePoseDraw page.

Test	**R**esult
TextToMotion link	Pass
Register Link	Pass
Sign up Link	Pass
Poses shown when TextToMotion is clicked	Pass
Search Link clicked	Pass
About Link clicked	Pass
ImagePoseDraw link clicked	Pass

3.1.2 Updating the database

Description

The database allows users to upload images to the database through the website interface. A test is considered a pass if an image or video can be uploaded, then accessed by user through the web interface.

Input Data

Input	Description
TomBrady.jpg	A picture of a man standing still and facing the camera
AverageGirl.jpg	A picture of a woman standing still and facing the camera
AverageGuy.jpg	A picture of football star Tom Brady

Results

It was possible to upload all of the test images to the database through the website interface.

Test	**R**esult
Tom Brady	Pass
Average Girl	Pass
Average Guy	Pass

3.1.3 Running Pose Estimation

Description

Once a user uploads an image they should be able to run pose estimation on an uploaded image. A test is considered a pass if an image or video can be uploaded, have pose estimation run on it, and have the result be visible to the user.

Results

It was possible to run pose estimation on the uploaded images, by submitting the images in a web form either through URL or from file.

The resultant poses showed up under the `ImagePoseDraw/Details/N` page for each upload.

Test	**R**esult
Tom Brady	Pass
Average Girl	Pass
Average Guy	Pass

3.1.4 Search by Tag or Name

Description

The database has the ability to be queried using a video tag or the video name. A test is considered a pass if we can search the database using a tag or name and the correct video appears.

Results

Test	**R**esult
Tom Brady	Pass
Average Girl	Pass
Average Guy	Pass

All of these tests passed. It was possible to search through the database and retrieve entries by tag and video name.

3.2 Solution Constraints Testing

3.2.1 Deep Learning Methods Test

Description

Our supervisor Dr. Taylor will check if the deep learning methods used are modern and up-to-date.

Results

Our supervisor confirmed that the deep learning methods used achieve near state-of-the-art results using modern methods, based on the paper of Bulat et al.

Test	**R**esult
Dr. Taylor	Pass

3.2.2 Standard Data Format Test

Description

An automated test that checks if the human pose data used for the project is standard and compatible with existing software libraries.

Results

This test was technically a failure. The data was not converted to a format compatible with libraries. Instead the data is stored in JSON strings, which is a standard format, but not compressed as an ideal format such as HDF5 would be.

4 Functional Requirements

4.1 Supported Video Encodings Test

4.1.1 Description

Tests whether the ReadFrames API is able to decode MP4 files. If we are able to run pose estimation on the video, then the ReadFrames API is able to process the frames.

4.1.2 Input Data

Input	Description
E9FY2.MP4	A short video of a woman eating a sandwich
U4XV9.MP4	A short video of a man waking up and getting out of bed
Z1A0Q.MP4	A short video of a man sitting on a stool

Results

Test	**R**esult
`E9FY2.MP4`	Pass
`U4XV9.MP4`	Pass
`Z1A0Q.MP4`	Pass

4.2 Frame Reading Timestamp Accuracy Test

4.2.1 Description

Tests whether the timestamps on the frames returned by the ReadFrames API match their temporal position in the original video stream. Our input data is identical to the previous test.

4.2.2 Results

Test	**R**esult
`E9FY2.MP4`	Pass
`U4XV9.MP4`	Pass
`Z1A0Q.MP4`	Pass

4.3 Human Pose Estimation Data Quality Test

4.3.1 Description

Test to ensure the data quality produced by the human pose estimator component was acceptable.

A set of Charades videos will be processed by the human pose estimator, and skeleton animations corresponding to the generated human pose data will be created (this is a scoped part of the software pipeline). A double-blind test will be ran, wherein testers will be shown random mixed sets of the skeleton animations produced by McMaster Text to Motion, together with skeletons from actual motion capture data coming from CMU's motion capture lab. Testers will indicate whether they think the motion capture data came from actual motion capture, or from the pose estimation software.

The McMaster Text to motion Results should be guessed as accurate at similar rates to the Charades tests.

4.3.2 Results

We provide the ratio of Text-to-Motion images guessed as accurate compared to the Charades Images.

Test	**C**harades/TextToMotion	**R**esult
Nick	1/1	Pass
Maddy	7/5	Pass
Sarah	5/4	Pass

4.4 Database Output Full Range Coverage Test

4.4.1 Description

Tests to be sure all entries in the database can be successfully searched for. The videos provided from earlier tests are put into the database, and have been renamed for testing purposes.

4.4.2 Input Data

Input	Description
`Waking_Up.mp4`	Given tags sleeping, boy, man, sleepy, getting up, table
`Eating.mp4`	Given tags sandwich, eating, girl, woman, table
`Stool.mp4`	Given tags man, stool, corner, sitting

Results

The given set of videos appeared in the returned list of videos from a database search.

Test	**R**esult
`Waking_Up.mp4`	Pass
`Eating.mp4`	Pass
`Stool.mp4`	Pass

4.5 Database No False Positives

4.5.1 Description

Tests that the database search does not return any false positives, such as videos or images that do not contain searched words. The same videos from the previous test will be used with the same tags. Thus, we will search with tags other than those provided. If no videos appear, then the test is a success.

4.5.2 Input Data

Input	Description
`Waking_Up.mp4`	Given tags luggage, chair, end, loading
`Eating.mp4`	Given tags recollection, breaking, band, insult
`Stool.mp4`	Given tags tackling, interview, virus, sunk

Results

Using the nonsensical keywords from our input data, no search results were returned, meaning that the test passed.

Test	**R**esult
Waking_Up.mp4	Pass
Eating.mp4	Pass
Stool.mp4	Pass

4.6 Full Text Search Order by Relevance Test

4.6.1 Description

Using the data from the previous test, we will conduct a search with multiple tags and the videos output by the search should should be ordered from the most relevant video to the least relevant.

4.6.2 Input Data

Input	Description
man, stool	should return Stool.mp4 followed by Waking_Up.mp4
man, bed	should return Waking_Up.mp4 followed by Stool.mp4
table, girl	should return Eating.mp4 followed by Waking_Up.mp4

Results

The search results returned the correct relative ordering for the videos.

Test	**R**esult
man, stool	Pass
man, bed	Pass
table, girl	Pass

5 Non-Functional Requirements

5.1 Usability

5.1.1 Description

In order to determine the usability of the Text-to-Motion database, a small sample of users were asked to use the website to perform some predetermined actions and answer questions afterwards.

Before the participants were asked to preform any actions they were given a minute to familiarize themselves with the interface, but were not given any guidance or tips from the development team.

Once the time was up they were asked to upload an image on mobile or desktop through their webcam, a URL or from a file saved within the computer.

While performing the required action the participant's time was recorded and used to determine if a requirement had passed or failed.

Upon completion of the task the users were asked to rate the style and design of the website on a scale from 1 to 10.

5.1.2 Results

The results from the participants can be seen throughout the Non-Functional Requirements along with a pass or fail based on the requirements description.

5.2 Look and Feel Requirements

5.2.1 Colour Scheme

Description

A test to see if the colour scheme of the website is visually appealing. The participants were asked to rate the websites colour scheme on a scale from 1-10, and any result above a 6 will be considered a pass.

Results

The users polled rated the visual appeal of the colour scheme high enough for this test to pass.

User	Rating	Result
Nick	7	Pass
Maddy	8	Pass
Sarah	7	Pass

5.3 Style Requirements

5.3.1 Minimalistic Web Design

Description

The website interface should be minimal and should inform the user of valid actions through visual means. Participants were asked to rate the design from 1-10, and any result above a 5 will be considered a pass.

Results

Participants rated the design of the website interface highly enough to warrant a pass.

User	Rating	Result
Nick	9	Pass
Maddy	8	Pass
Sarah	8	Pass

5.4 Ease of Use Requirements

5.4.1 Upload/Download

Description

Through the web interface a user should be able to upload a picture using either a mobile phone camera, URL, or saved file. The participant will start on the Home page and be asked to upload an image through one of the methods just mentioned. In order for this to be considered a pass it should take the users 30 seconds or less to complete the upload process and click the button.

Results

All users were able to upload images within the required time, and therefore the tests passed.

Test	User	Time	Result
Uploading based on URL	Nick	28 seconds	Pass
Uploading image from a mobile device	Maddy	22 seconds	Pass
Uploading file from Desktop	Sarah	26 seconds	Pass

5.4.2 Text Box Functionality

Description

The user should be able to input a descriptive word or phrase into a text-box from within the web interface in order to search for a video. In order to complete this task the users were asked to search for a specific word and display the results. Any time below 10 seconds will be considered a pass.

Results

Test	User	Time	Result
Search for "Woman"	Nick	4 seconds	Pass
Search for "f"	Maddy	3 seconds	Pass
Search for "the"	Sarah	4 seconds	Pass

5.5 Learning Requirements

5.5.1 Usability Tests

Description

The user should be able to interact with the website without prior knowledge. They will given a minute to explore the website. After that time the participants were asked to rate the usability on a scale of 1-10. An average of 6 is required for a pass.

Results

The users' usability ratings allowed the site to pass this test.

User	Rank	Result
Nick	6	Pass
Maddy	8	Pass
Sarah	8	Pass

5.6 Politeness and Understandability Requirements

5.6.1 Hiding the Inner Workings

Description

Users should not be able to see the deep learning model and its training when using the pose estimation. When prompted the website should display the correct skeletons without any low-level detail. Once uploaded the participants were asked if they saw anything that seemed out of place or any information on the deep learning process, if they did not it will be considered a pass.

Results

Users indicated that the deep learning model was encapsulated from their view, and hence this test passed.

Test	**U**ser	**R**esult
Uploading an image from URL	Nick	Pass
Uploading an image from mobile	Maddy	Pass
Uploading an image from desktop	Sarah	Pass

5.7 Speed and Latency Testing

5.7.1 External Database Connection Response Time

Description

The web interface should be able to connect to an external database and store or query items. In order for this test to be considered a pass the confirmation of the image being uploaded would have to occur within 30 seconds so that additional resources are not wasted by the database. Testing this will occur by uploading an image and testing the total time taken.

Test	**T**ime	**R**esult
Uploading the image from a URL	28 seconds	Pass
Uploading an image from desktop	29 seconds	Pass

Input Data

Input	Description
Image from a URL	An image of a male
Image that was saved within the desktop	An image of Seth Rogan

Results

According to the response times of this automated test, the database queries were executed fast enough for the test to pass.

5.7.2 Website Search Responsiveness

Description

When given a word or phrase the web interface will be able to respond with an image or video of a pose or action within a two minutes.

Input Data

Input	Description
Creepy	Searched using a tag within the description
Seth Rogan	Searched using a tag within the description

Results

Test	**T**ime	**R**esult
Search for “Seth”	1 second	Pass
Search for “Creepy”	2 seconds	Pass

6 Other Relevant Testing

Again we test this on the same files we have used in the previous tests: `TomBrady.jpg`, `AverageGirl.jpg` and `AverageGuy.jpg`.

6.1 Precision and Accuracy

6.1.1 Bone and Joint Position

Description

The pose estimation should accurately predict the placement of joints and bones of the person in the provided photo. This will be determined with visual means with an uncertainty range of 20 pixels.

Input Data

Input	Description
`TomBrady.jpg`	A picture of a man standing still and facing the camera
`AverageGirl.jpg`	A picture of a woman standing still and facing the camera
`TomBrady.jpg`	A picture of football star Tom Brady

Results

We were able to qualitatively confirm that these tests passed for the given input images. A more rigorous "PCKh" metric is used to formally evaluate the performance of our deep learning model on single-person pose estimation.

Test	**R**esult
Tom Brady	Pass
Average Girl	Pass
Average Guy	Pass

6.1.2 Deep Learning Model

Description

The **PCKh** metric, used by the MPII Human Pose Dataset, defines a joint estimate as matching the ground truth if the estimate lies within 50% of the head segment length. The head segment length is defined as the diagonal across the annotated head rectangle in the MPII data, multiplied by a factor of 0.6. Details can be found by examining the MATLAB evaluation script provided with the MPII dataset.

If our model can achieve 80% total PCKh then the test is considered a pass.

Results

Our model achieves 85% PCKh, thus this test is a pass.

Test	**PCKh Value**
r ankle	68.907%
r knee	77.201%
r hip	83.583%
l hip	84.444%
l knee	77.419%
l ankle	69.055%
pelvis	89.776%
thorax	98.071%
upper neck	97.823%
head top	96.557%
r wrist	81.288%
r elbow	87.205%
r shoulder	92.918%
l shoulder	92.828%
l elbow	85.278%
l wrist	80.719%
Total PCKh	**85.195%**

6.2 Reliability and Availability Requirements

6.2.1 Software Availability

Description

The software component of the project should be available at all times. If we can have an event regularly occur then it will be considered a pass. To do this we arranged to have the pose estimation algorithm automatically called on to process a single image at 4 hour intervals, and record the time.

Results

The software successfully processed the image at the specified 4 hour intervals over a period of 2 days.

6.2.2 Website Availability

Description

The software component of the project should be available at all times with the exception of maintenance and migration. When we make a web server call we should receive an HTTP verified response. To do this we will have three users sending a HTTP POST request to the server. If they receive a response the test is considered a pass.

Results

All three users were able to receive responses to their HTTP POST requests, therefore we consider this test a pass.

6.3 Robustness or Fault-Tolerance Requirements

6.3.1 Web Interface Error Handling

Description

The web interface should respond to unhandled exceptions by throwing the corresponding error messages. If an exception is thrown and an error message is displayed then the test is considered a pass.

Input Data

Input	Description
Scenario 1	Try uploading an image while the image storage service (Amazon S3) is not available.
Scenario 2	Trying to search an image on the database while the database server is down.

Results

For Scenarios 1 and 2, appropriate exceptions were thrown and error pages displayed to the user. Therefore this test is considered to have passed.

6.4 Capacity Requirements

6.4.1 Multiple Connections

Description

The web interface should be able to serve multiple connections. If the interface can support 5 connections at once it is considered a pass.

Input Data

Input	Description
Andrew	The connection to the website corresponding to tester Andrew
Brendan	The connection to the website corresponding to tester Brendan
Udip	The connection to the website corresponding to tester Udip
Jordan	The connection to the website corresponding to tester Jordan
Dave	The connection to the website corresponding to tester Dave

Results

The 5 users were able to successfully connect to the website, and run queries simultaneously without experiencing significant slow down or waiting.

6.4.2 Database Capacity

Description

The database should contain at least 5GB of data in order to facilitate growth.

Results

The database successfully handled us uploading a total of 5.1GB of data in the form of pictures of various qualities. These pictures were of randomly acquired photos from Google, which were combined to form the data used in this testing.

6.5 Scaling of Extensibility Requirements

6.6 Operational Environment Requirements

6.6.1 Linux Friendly TensorFlow

Description

The web interface should be run on a Linux friendly server that can access the TensorFlow model either directly or indirectly. By creating an interface that successfully runs on a Apache or nginx server this test will be considered a pass.

Results

Our service is successfully running on a production Linux server using nginx.

6.6.2 Export Types

Description

The project should be able to export multiple types of media (JPEG, PNG, etc) in order to support all major operating systems. We will use `TomBrady.jpg` for this test.

Results

This test was a guaranteed pass due to the fact that our website stores images as a base64, meaning it can be converted into any type.

Test	**R**esult
PNG	Pass
JPEG	Pass
BMP	Pass

7 Traceability

Table 7.1: Traceability Matrix for Test-Requirement Relationships

Test	Description	Requirement
3.2, 4.2, 4.3, 5.8.2	Tests which measured performance Accuracy of Deep Learning Algorithm	Req 1, 8, 23(Speed and Latency)
4.1, 5.3.1, 5.7.1, 6.4.2	System tests, measuring reliability of the web framework.	Req 7, 12, 17, 20, 30, 38, 39
4.4-4.6	Unit and Systems Tests Grading Database Search Performance	Req # 9, 10, 11,
4.5, 6.3.1, 6.3.1, 6.4.1	Security and Data Integrity Tests	Req # 10, 27, 29
4.1.1, 5.8.2, 6.4.1	Proper Formatting Tests	Req # 7, 23, 29
4.2.1-5.8.1,6.1.1-6.2.2	User Interface Tests	Req # 13-21, 24-26

7.1 Modules

Similarly, the following is a traceability table explicitly relating test cases to modules:

Table 7.2: Tests and Modules Relationships

Test #	Module
3.1(all subsections), 3.2.1, 4.4-4.6, 5(all sections), 6.2(all subsections) - 6.4(all subsections), 6.6.2	ASP.NET and DB
3.1.3, 4.3, 5.8.2, 6.1, 6.5	TensorFlow Models
3.13, 3.2.2, 4.1-4.2, 5.8.2, 6.2(all subsections), 6.6.1	Python HTTP Server

8 Changes After testing

The first of our major changes would likely be to the website interface. Though our testers reviewed it favourably – there were numerous references – but we were also given consistent criticism that an alternate colour scheme might be a slight improvement. As such we have agreed to experiment with those improvements.

Testing also revealed that a JavaScript application to allow continuous requests to the HTTP server as well as utilize a mobile platform would have greater applicability to the Guelph team. This improvement would come alongside a function on the website to display statistics. This would help insure greater availability. The ability to track the database live would be both useful to the testing team as well as an entertaining aspect for the product demo.

The last major improvement we wish to provide is an improved JavaScript plugin for drawing the skeleton on top of an image. Testing revealed that there was flicker with the drawn skeletons, and as such, could lead to false negatives, where a skeleton that appears inaccurate is actually just limited by our current skeleton drawing methods.