Test Summary Report

Contents

[**1.** **Purpose** 2](#_heading=h.gjdgxs)

[**2. Application Overview** 2](#_heading=h.30j0zll)

[**3. Testing Scope** 2](#_heading=h.1fob9te)

[**4. Metrics** 2](#_heading=h.3znysh7)

[**5. Types of testing performed** 3](#_heading=h.2et92p0)

[**6. Test Environment & Tools** 3](#_heading=h.tyjcwt)

[**7. Lessons Learnt** 3](#_heading=h.3dy6vkm)

[**8. Recommendations** 3](#_heading=h.1t3h5sf)

[**9. Best Practices** 3](#_heading=h.4d34og8)

[**10.**](#_heading=h.2s8eyo1) **Exit Criteria** 3

[**11.**](#_heading=h.17dp8vu) **Conclusion/Sign Off** 4

[**12.**](#_heading=h.3rdcrjn) **Definitions/Acronyms/Abbreviations** 4

# **Purpose**

This document explains the various activities performed as part of testing of team Arti-Gan’s applications

# **2. Application Overview**

Team Arti-Gan’s project application is a GaN which takes in a dataset of images to train a model which then produces a new image every time the model progresses. A website is used to present the project as well and to access these images produced by the model.

# **3. Testing Scope**

1. In Scope

Functional testing for the output of the image produced by the GaN is in right specifications

Functional testing for website being served and all web pages are accessible

1. Out of Scope

Performance of training the GaN was not done for this application

Mobile Accessibility was not done for this application website

1. Items not tested

Serving of the website on a third-party server was not done and can be tested on final production of the applications

# **4. Metrics**

1. No. of test cases planned vs executed/No. of test cases passed/failed

| Test cases planned | Test cases executed | TCs Pass | TCs Failed |
| --- | --- | --- | --- |
| 6 | 6 | 6 | 0 |

1. No. of defects identified and their status/severity

|  | Critical | Major | Medium | Cosmetic | Total |
| --- | --- | --- | --- | --- | --- |
| Closed | 4 | 0 | 0 | 0 | 4 |
| Open | 0 | 0 | 0 | 2 | 2 |
|  |  |  |  |  | 6 |

1. Defects distributions

|  | GaN | Website | **Total** |
| --- | --- | --- | --- |
| Critical | 4 | 0 | **0** |
| Major | 0 | 0 | **0** |
| Medium | 0 | 0 | **0** |
| Cosmetic | 0 | 2 | **2** |
| **Total🡪** | **4** | **2** | **6** |

# **5. Types of testing performed**

1. Regression Testing

* Regression testing was performed on the website each time a new build was deployed that contains fixes to defects and all pages were accessible

# **6. Test Environment & Tools**

| Website URL (done locally) | localhost:port |
| --- | --- |
| Web-server | Node.js |

# **7. Lessons Learnt**

| Issues faced | Solutions |
| --- | --- |
| Can’t tell what the model was doing so issues could not be fixed. | Assert at each layer to have a better understanding of what was happening. |
|  |  |

# **8. Recommendations**

When developing a GaN, always test to see if layers are being changed correctly.

# **9. Best Practices**

* With the website being served on node.js, testing of the accessibility of the website could be done proactively including the finding and solving of defects.
* By using python, we were able to use assert to make sure that the shape of the images were being correctly changed at each layer of the neural network.

# **10. Exit Criteria**

1. All test cases should be executed – **Yes**
2. All defects in Critical, Major, Medium severity should be verified and closed - **Yes**
3. Any open defects in trivial severity – **Planned actions for defects with website are prepared within expected due dates**

# **11. Conclusion/Sign Off**

The team has determined that the application is good to go as it meets the Exit Criteria and planned actions for defects are in place.

# **12. Definitions/Acronyms/Abbreviations**

* GaN- Generative Adversarial Neural Network
* No.- Number

Arti-Gan Team – Nathanael Mann, Mark Bridgewater, Joshua Carrier