

Faculty for Computer Science, Electrical Engineering and Mathematics  
Paderborn University  
Communications Engineering  
Prof. Dr.-Ing. Reinhold Häb-Umbach

## **Proposal for Masterarbeit**

### **Detecting Fake Images**

by

Akshit Bhatia  
Matr.-No.: 6868664

Supervisor: Philipp Terhoerst  
Filing date: 31.02.2042  
Number: EIM/NT – Sommersemester 2023

---

# Declaration

---

This Proposal for Masterarbeit, with the title “Detecting Fake Images”, is the result of my own work and includes nothing that is the outcome of work done in collaboration, except where specifically indicated. It has not been published yet or submitted, in whole or in part, for a degree at any other university.

Paderborn, 31.02.2042

---

(Akshit Bhatia)

---

# Abstract

---

---

# Contents

---

|  |           |
|--|-----------|
| Declaration  | ii        |
| <b>Abstract</b>                                      | iii       |
| <b>1 Introduction</b>                                | <b>1</b>  |
| <b>2 Related Work</b>                                | <b>2</b>  |
| <b>3 Data Collection</b>                             | <b>3</b>  |
| 3.1 Fake Data . . . . .                              | 3         |
| 3.2 Real Data . . . . .                              | 3         |
| <b>4 Thesis Goal</b>                                 | <b>4</b>  |
| <b>5 Outline of Topics in the Final Thesis</b>       | <b>5</b>  |
| <b>6 Work Plan</b>                                   | <b>6</b>  |
| <b>A Appendix</b>                                    | <b>7</b>  |
| A.1 Kompilieren . . . . .                            | 7         |
| A.2 Textbezogene Befehle . . . . .                   | 7         |
| A.3 Einheiten mit speziellem Paket setzen . . . . .  | 7         |
| A.4 Mathematik . . . . .                             | 8         |
| A.5 Abkürzungen . . . . .                            | 8         |
| A.6 Pseudo-Code . . . . .                            | 8         |
| A.7 Grafiken . . . . .                               | 8         |
| A.8 Blockdiagramme . . . . .                         | 10        |
| A.9 Tabellen . . . . .                               | 11        |
| A.10 Hinweise zum Drucken der Ausarbeitung . . . . . | 11        |
| <b>List of symbols</b>                               | <b>12</b> |
| <b>List of Figures</b>                               | <b>13</b> |
| <b>List of Tables</b>                                | <b>14</b> |
| <b>Acronyms</b>                                      | <b>15</b> |
| <b>Bibliography</b>                                  | <b>16</b> |

---

# 1 Introduction

---

In the painting exhibitions, we see many paintings hanging on the wall along with a textual description. An artist would have taken hours to paint such a painting but he would take only few minutes to describe his work in words. Generative AI is a paradigm in Artificial Intelligence field that provides text-to-image models. These models can create an image when only a textual description of the image is provided by the user. These models are capable to create an image that captures most of the given description and they usually produce a realistic image full of creativity and imagination. However, with such an improvement over image generation using AI, hackers can generate synthetic images to fool the security systems. One business where this malpractice can have a major impact is the business of the insurance companies. Because these text-to-image models have been trained to understand on large number of topics, hence the gap between a fake image and a real image is diminishing.

Consider a liability insurance company. This company receives claims that include images of broken infrastructure in a house. It could be a broken window, or a broken wall, or a fence destroyed because of a natural calamity like floods. The company verifies the claim's authenticity and makes a decision. This decision will either be to reimburse the repairing cost of the damage or to reject the claim. Rejecting the claim would mean that no reimbursement is provided to the customer. For this company, passing a false claim would mean loosing the business. But for the same company, rejecting an authentic claim wouldn't have as big impact as for the other case. If a well written textual prompt is given to a text-to-image models, one can synthesize an image of a damaged infrastructure such that it will come under the liability insurance scope.

Through this thesis topic, **we propose** to build a classifier that identifies if an image is synthetically generated or not. More specifically, the thesis topic focuses on the images of damaged parts of the house. Thus an image of a broken window or an image of a broken pipe will be perfect candidates for the classifier to be trained on. The original images are provided by **PropertyExpert**, a company that works with different insurance companies and verifies the claims that these insurance companies receives. [HDHU17]

---

## 2 Related Work

---

To the best of our knowledge, the work [1 -i De-Fake] is most similar to this work. In the work [1], the authors have set up three research questions (RQ). Their RQ 1 is to build up a classifier that can differentiate between fake images and real images. Their RQ 2 is to find the sources of the fake image that the classifier identify and their RQ 3 is mapping the quality of fake image generation with textual prompts. The RQ1 which is to build a classifier that identifies if the image is fake or not, is same as ours. To accomplish this goal, they have trained a classifier on two different datasets, one extracted from COCO dataset and the other extracted from Flickr30k dataset. To train the classifier, the authors adopted two different strategies and compared their results. In Approach 1 (Image only with Resnet18 based classifier), they trained the classifier only on the images from the datasets they have collected. In Approach 2 (Image and given prompt with MLP based classifier), to every image the authors had a prompt attached. Then the authors used CLIP's image encoder and text encoder to create embeddings of the image and the prompt. They combined both the embeddings and trained a 2-layered Multilayered Perceptron on these embeddings. For the prompts, they had two different strategies. One being using the natural prompts (given in the dataset), other being generating the prompts using BLIP model [BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation].

---

## 3 Data Collection

---

For this thesis, the resulting classifier should be able to identify the images of the damaged furniture of a house as a real image or a fake image.

### 3.1 Fake Data

We will use different tools to create fake images. We will use stable diffusion to create majority of fake images because stable diffusion is an open source platform and hence one can create many images without incurring any cost. We will also use DALL-E2, Midjourney and Adobe Firefly to create some images. But these images will serve as test dataset mainly because these platforms are not free to use. Besides this, we also have an access to thousands of images from COCO dataset. These images are based on general topics like airplanes, cars etc. COCO dataset has fake images generated on different tools like Midjourney, stable diffusion etc.

### 3.2 Real Data

The company Propertyexpert has agreed to share the real data. The company receives insurance claims from different insurance companies and the company has hired several craftsmen who verifies the claim and determine whether the claim must be accepted or rejected. For our purpose, the company has shared images of broken furniture of the house. These photos are taken by the users with the help of their personal cameras and also attached a prompt that describes the image. These images are only accessible from the company's local servers and will not be shared to anyone who is not part of this company. Other than this, the COCO dataset which is publicly available also has real images.

---

## 4 Thesis Goal

---

With this thesis, we aim to develop a classifier that can with a higher and reliable accuracy detect fake images. This classifier would detect the presence of tampering or noise in an image. Different model and combinations of techniques will be compared against eachother. The following are the interesting questions that this thesis will explore and answer:

1. If the model is trained on general images from COCO dataset as in the work [De-fake], will the accuracy for the classifier be the same as the paper [De-fake] claims when the testset will have data related to broken furniture of the house?
2. What would be the performance of the model if it is trained and tested on data related to only the broken furniture of the house?
3. Will the performance of the model improve when every image is analysed using Error Level Analysis? Error Level Analysis identifies the parts of the image that have different compression levels. It analyses compression artifacts in the image [wikipedia definition].
4. Combine the artifact analysis in an image from paper [Intriguing Properties of synthetic images by NVIDIA] and check the performance of the model.



1. select very good images and get them verified and get them tested on the netwrok
2. print and scan
- 3.



---

# 5 Outline of Topics in the Final Thesis

---

1. Abstract
2. Introduction
3. Related Work
4. Fake Image Generation using Generative AI
5. Identification of Fake Images using Artifacts
6. Diffusion Models that reduces the presence of Artifacts in the image
7. Basic functioning of Stable Diffusion and DALLE2
8. CLIP's functionality
9. BLIP to Generate prompts
10. Experiment Details 
11. Results
12. References 

# 6 Work Plan

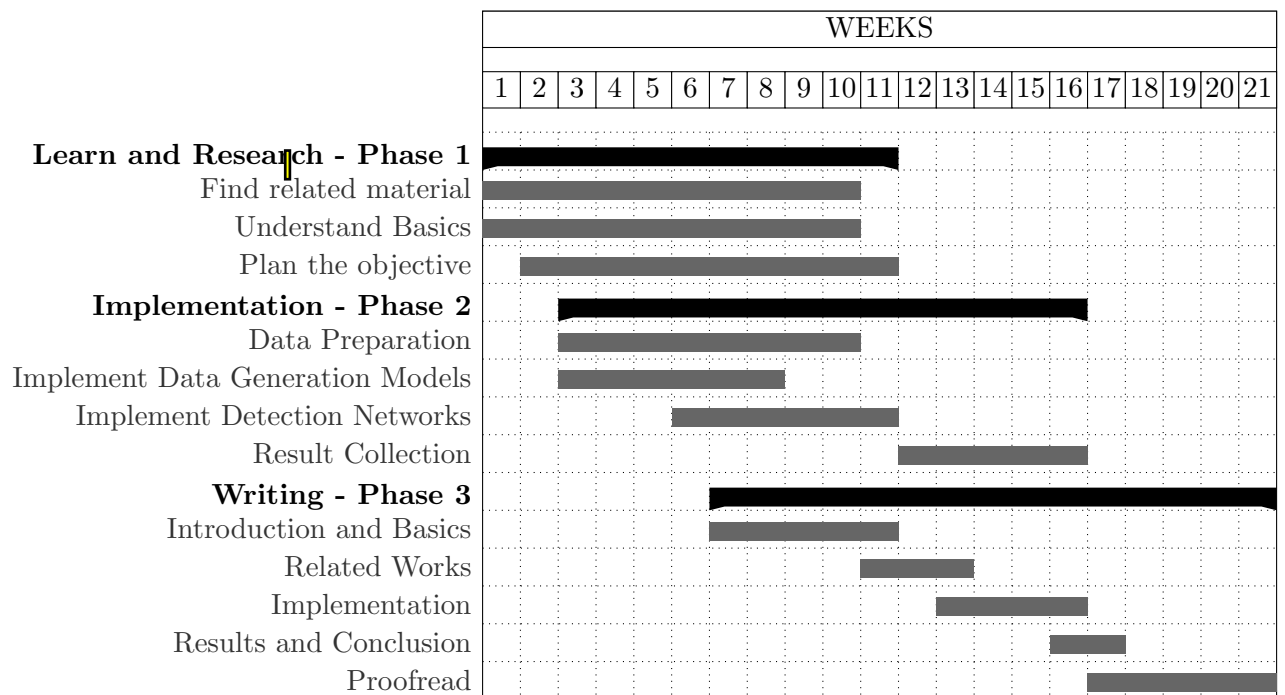


Figure 6.1: Work Plan

---

# A Appendix

---

Längere Herleitungen können im Anhang untergebracht werden. In diesem Fall sind die wichtigsten  $\text{\LaTeX}$ -Beispiele zusammengefasst.

## A.1 Kompilieren

Das Dokument kann mit folgendem Befehl kompiliert werden:

`make`

Mit folgendem Befehl werden die erstellten Dateien gelöscht:

`make clean`

Mit folgendem Befehl werden die Macros aktualisiert:

`make macros`

## A.2 Textbezogene Befehle

Noch zu bearbeitende Korrekturen können mit dem Befehl `\inred{Test}` in **rot** gesetzt werden.

URLs können mit `\url{http://www.very-very-long-url.de}` eingefügt werden.

Dies ist ein Beispiel: `http://tex.stackexchange.com/questions/49788/hyperref-url-long-url-with-dashes-wont-break`

Für kurze URLs `http://nt.upb.de`

## A.3 Einheiten mit speziellem Paket setzen

Im folgenden sind ein paar Beispiele zum Verwenden von Einheiten und Dezimalzahlen angegeben:

- Einheit ist durch halbes Leerzeichen getrennt: 10 km
- Spezielle Einheiten sind bereits vordefiniert:  $1 \times 10^6 \text{ }^\circ\text{C}$
- Es kann ein Punkt im Code verwendet werden: `0comma1 s`
- Ein Komma im Code ist auch möglich: `0comma001 s`
- Dezimalzahlen allen sollten auch mit diesem Paket gesetzt werden: `1comma0`
- Einheiten allein ebenfalls: km

## A.4 Mathematik

Im Text können Formeln mit den Befehlen `\alpha` `\neq` `\vect` `\alpha` eingefügt werden, das Ergebnis ist dann  $\alpha \neq \mathbf{f}$ . Am besten liest man sich `macros.tex` im Hauptverzeichnis durch um einen Überblick über alle Befehle zu bekommen.

Beispiele für nummerierte Gleichungen sind hier angegeben:

$$R(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} \tilde{x}(n) \tilde{x}(n+k) \quad k = 0, \dots, N-1 \quad (\text{A.1})$$

$$= \frac{R(k)}{R(0)}. \quad (\text{A.2})$$

Fallunterscheidung können wie folgt geschrieben werden.

$$x(n) = \begin{cases} e^{-j\omega Tn} & \text{falls } n > 0, \\ 0 & \text{sonst.} \end{cases} \quad (\text{A.3})$$

Matrizen und Vektoren sind fett zu setzen. Es wird `\vect X` aus `macros.tex` im Hauptverzeichnis empfohlen.

$$\mathbf{X} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (\text{A.4})$$

## A.5 Abkürzungen

Abkürzungen können in der Datei `chapters/glossaries.tex` definiert werden. Bei der ersten Verwendung einer Abkürzung wird diese automatisch definiert: Fast Fourier Transformation (FFT)

Bei jeder weiteren Verwendung wird nur noch die Abkürzung angezeigt: FFT

Sollen eigene Makros definiert werden, sind einige bereits in `chapters/macros.tex` zu finden.

## A.6 Pseudo-Code

Pseudo-Code sollte in fast allen Fällen dem Programm-Code vorgezogen werden. Ein Beispiel ist durch Algorithmus 1 gegeben.

## A.7 Grafiken

Matlab-Grafiken können mittels `matlab2tikz()` in tikz-Code exportiert werden. Dieser tikz-Code kann wie in Abbildung A.1 eingebunden werden. Ein ähnliches Paket für Python stellt `matplotlib2tikz` dar.

Ein Beispiel-Matlab-Skript befindet sich in `tikz/pdf.m`.

Es ist auch möglich, mehrere Bilder nebeneinander wie in Abbildung A.2 darzustellen.

---

**Algorithm 1** Offline source counting algorithm
 

---

- 1: Calculate  $A^{(1)}(t, f)$
  - 2: **for**  $\nu = 1, \dots, \nu_{\max}$  **do**
  - 3:   Use VEM algorithm
  - 4:   Calculate principal component  $\mathbf{W}_\nu = \mathcal{P}(\mathbf{B}_\nu)$
  - 5:   **if**  $\nu < \nu_{\max}$ : **then** Reweight observations **end if**
  - 6: **end for**
  - 7: Count iterations where  $\kappa_\nu > \kappa_{\text{Th}} \wedge s_\nu < s_{\text{Th}}$
- 

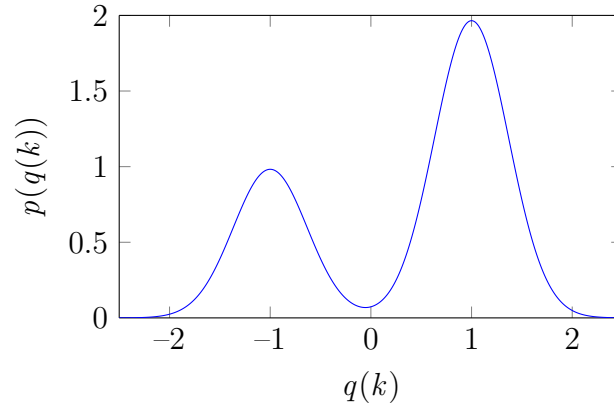


Figure A.1: Beispielbild

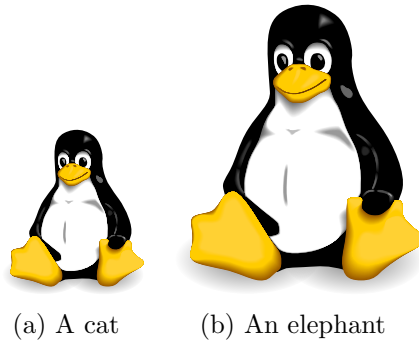


Figure A.2: Two animals

## A.8 Blockdiagramme

Blockdiagramme können mittels Inkscape erstellt werden. Die Diagramme bitte immer im Format svg speichern, um spätere Änderungen zu ermöglichen. Anschließend können die Bilder mittels "Speichern unter" als "Portable Document Format \*.pdf" exportiert werden. Bitte verwendet die folgenden Einstellungen:

- PDF-Version: PDF 1.5
- Einstellungen für Textausgaben: Texte in PDF weglassen und LaTeX Datei erstellen
- Filtereffekt in Raster umwandeln: Aktiv
- Auflösung des Rasters (dpi): 96
- Seitengröße der Ausgabe: Größe des exportierten Objekts verwenden
- Beschnittzugabe (mm): 0,0
- Export beschränkt auf das Objekt mit ID: *leer lassen*

Im Text das Bild mittels

```
\begin{figure}[htb]
\centering
\def\svgwidth{0.99\columnwidth}
\import{images/}{task.pdf_tex}
\caption{Wireless acoustic sensor network.}
\label{fig:task}
\end{figure}
```

einbinden. Wichtig ist hierbei den Befehl "import" zu verwenden und im ersten Argument den Pfad zum Bild anzugeben (hier: images, immer gefolgt von einem Slash). Sollen Textteile ergänzt oder ersetzt werden, so kann dies in der zugehörigen tex Datei (hier: task.pdf\_tex) erfolgen.

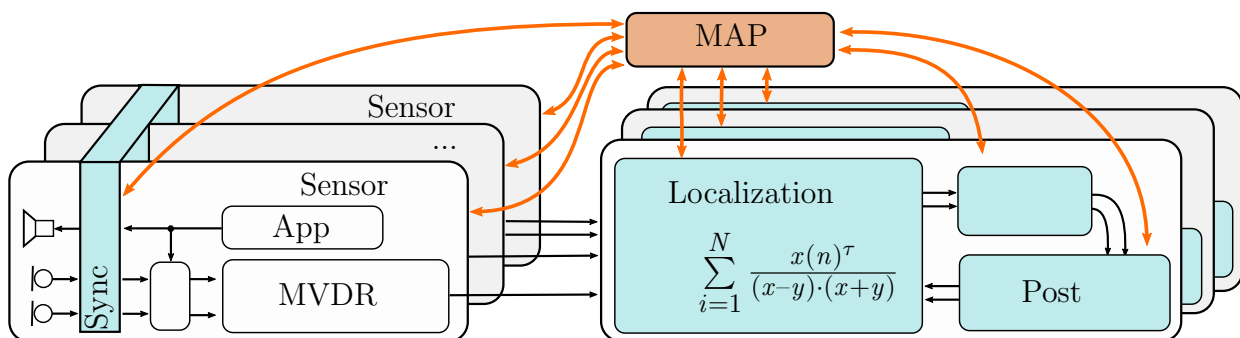


Figure A.3: Wireless acoustic sensor network.

Table A.1: Tabellen besitzen Tabellenüberschriften im Gegensatz zu Bildern

| Head 1 | Head 2 |
|--------|--------|
| a      | b      |
| d      | e      |

## A.9 Tabellen

Eine Tabelle gemäß des Chicago Manual of Style [Wil+10] ist Tabelle A.1.

## A.10 Hinweise zum Drucken der Ausarbeitung

In der Regel wird die Ausarbeitung einseitig auf 80 g Papier DIN A4 Format gedruckt.

Vor dem Binden der Ausarbeitung sollte man ein leeres Blatt sowohl vor dem Titelblatt als auch nach der letzten Seite manuell hinzufügen.

Für die Bindung wird eine Hard-Cover Magister DIN A4 Format in Farbe weinrot empfohlen.

---

# List of symbols

---

|                       |                              |
|-----------------------|------------------------------|
| $\varphi_{xx}(t)$     | Autokorrelationsfunktion     |
| $\Phi_{xx}(j\omega)$  | Leistungsdichtespektrum      |
| $\text{tr}(\cdot)$    | Spur-Operator                |
| $\mathbb{E}\{\cdot\}$ | Erwartungswert-Operator      |
| const.                | Additive beliebige Konstante |
| j                     | Komplexe Einheit             |
| e                     | Eulersche Zahl               |



---

# List of Figures

---

|     |   |    |
|-----|---|----|
| 6.1 | Work Plan . . . . .                       | 6  |
| A.1 | Beispielbild . . . . .                    | 9  |
| A.2 | Two animals . . . . .                     | 9  |
| A.3 | Wireless acoustic sensor network. . . . . | 10 |

---

# List of Tables

---

|     |                                 |    |
|-----|---------------------------------|----|
| A.1 | Tabelle mit Kurztitel . . . . . | 11 |
|-----|---------------------------------|----|

---

# Acronyms

---

**FFT** Fast Fourier Transformation.

---

# Bibliography

---

- [HDHU17] J. Heymann, L. Drude, and R. Haeb-Umbach. “A Generic Neural Acoustic Beamforming Architecture for Robust Multi-Channel Speech Processing”. In: *Computer Speech and Language* (2017). URL: [http://nt.uni-paderborn.de/public/pubs/2017/ComputerSpeechLanguage\\_2017\\_heyman\\_paper.pdf](http://nt.uni-paderborn.de/public/pubs/2017/ComputerSpeechLanguage_2017_heyman_paper.pdf).
- [Wil+10] J. M. Williams et al. “Chicago Manual of Style”. In: (2010).