This is the Title of your very Awesome Thesis It's long to test line breaking

Masterarbeit

Firstname Surname

Abteilung Translationale Chirurgische Onkologie
NCT Dresden

Erstgutachter:	Prof. Dr.-Ing. Stefanie Speidel
Zweitgutachter:	xxxxxx
Betreuender Mitarbeiter:	xxxxxx

Bearbeitungszeit: December 19, 20XX – June 19, 20XX

Ich versichere hiermit, die vorliegende Arbeit selbstständig angefertigt zu haben. Die verwendeten Hilfsmittel und Quellen sind im Literaturverzeichnis vollständig aufgeführt.

Dresden, den YY Monat, 20XX

Zusammenfassung

Short description of motivation, methods, results and discussion

Inhaltsverzeichnis

1. Readme

As a placeholder, here are some hints for writing. You can remove this chapter when writing your thesis (remove the line `\include{chapters/readme}` in thesis.tex).

1.1 Getting Started

- First, decide on the language - if you want to write in German, comment out the line '`\selectlanguage{english}`' in thesis.tex.
- We have Texmaker installed on most machines, but you can also use a Tex editor of your choice.

1.2 Figures and Images

- If possible, always use vector graphics.
- Recommended image formats: PDF, EPS, SVG
- If you want, you can use the Latex TIKZ package to draw beautiful, scaling images or graphs.
- Make sure that there is at least one reference to each figure and table somewhere in the text (use `\ref{}`).

1.3 Texmaker

You can use any Tex editor and compiler you want to, but here are some hints for Texmaker, which is installed on our system:

- Open thesis.tex and go to *Options->Define Current Document as 'Master Document'*. This will always compile the full document (even if you're editing a subchapter).
- Go to *Options->Configure Texmaker->Quick Build* to choose what the Quick Build command (F1) should do.

1.4 General Hints

- Cite using the `\cite{}` Latex command, and put the corresponding bibliography files into *bibliography.bbl*.
  Example: The Dijkstra Algorithm [1] can be used to find the shortest path between two nodes in a graph.
- You can emphasize important names in-line by using the `\emph{}` command.
- Introduce abbreviations when you first use them and consistently use them in the remainder of the text.
- Try to be done a few days before the deadline so that your supervisor gets a chance to proofread before you hand it in.

2. Einführung

This section should describe your motivation and general introduction to the topic as well as the goals of your work.

2.1 Motivation

2.2 Ziele

2.3 ...

3. Theoretischer Hintergrund

3.1 Anfälligkeit gegenüber Störbildern

Die Anfälligkeit von Neuronalen Netzen gegenüber gezielt manipulierten Störbildern wurde erstmals 2013 von Szegedy et al. untersucht und auch wenn die genauen Hintergründe für diese Schwachstelle noch lang ungeklärt blieben, lässt sie sich heute genauer erklären. Um neuronale Netze möglichst effektiv trainieren und optimieren zu können, neigt man in der Regel dazu, ihr Verhalten während des Trainings möglichst linear zu halten. Selbst bei der Verwendung von vergleichsweise nichtlinearen Aktivierungsfunktionen, wie sigmoid oder softmax, ist man in der Regel bestrebt, eine Sättigung zu vermeiden und sich im quasi-linearen Mittelteil der Funktionen zu bewegen. Die führt dazu, dass die Netze sehr große Gradienten bezüglich der input-Werte bilden. Während dies zum einen natürlich ein effektiveres Lernen ermöglicht, bedeutet dies ebenfalls, dass diese Netze auf sehr geringe Änderungen der input-Werte mit sehr großen Änderungen der output-Werte reagieren. Dies wiederum bedeutet, dass lediglich wenige, oft für das menschliche Auge sogar unsichtbare, Manipulationen von Bildern nötig sind, um das Verhalten des Netzes auf diese Bilder grundsätzlich zu verändern. Dieser Zusammenhang tritt noch stärker zu Tage, umso größer die Auflösung der inputs ist, da der Manipulierende dadurch einfach mehr (und für die menschliche Wahrnehmung dadurch wesentlich weniger einflussreiche) Bildpunkte zu Verfügung hat, um seine gewünschte Reaktion zu erreichen und bei Bedarf zu verbergen.

3.2 Angriffsmöglichkeiten

Basierend auf den Hintergründen dieser Anfälligkeit, lassen sich verschiedene Angriffsmöglichkeiten formulieren, mit denen sich Störbilder für ein gegebenes Modell eines Neuronalen Netzes erstellen lassen. Die Formeln zu den folgenden Angriffen folgen weitestgehend der von Karakin et al. eingeführten Nomenklatur:

- X - Ein Eingabebild, also i.d.R. ein dreidimensionaler Tensor
- y_{true} - Die "wahre"Klasse des Eingabebilds, also die Reaktion des Netzes auf das nicht-manipulierte Bild
- $J(X, y)$ - Das Cross-Entropy-Loss des Netzes bei gegebenem Bild X und output y
- $Clip_{X,\epsilon}\{X'\}$ - Eine Funktion, die ein pixelweises Clipping des Bildes X' durchführt, sodass die Werte maximal um ϵ vom Original X abweichen

3.2.1 FGSM

Eine der ersten und noch immer populärsten Wege Störbilder zu generieren nennt sich FGSM - Fast Gradient Sign Method. Diese Methode wurde bereits 2014 von Goodfellow et al. vorgestellt und funktioniert folgendermaßen: Anstatt die berechneten Gradienten bezüglich eines inputs zu dazu zu verwenden, die Gewichte des Netzes zu verändern und ein möglichst niedriges loss zu erreichen, wird der input verändert, um ein möglichst hohes loss zu bekommen und somit eine Fehlklassifizierung zu erwirken.

$$X^{adv} = X + \epsilon sign(\nabla_X J(X, y_{true})) \tag{3.1}$$

3.2.2 Iterative Methode

Die von Karakin et al. eingeführte iterative Methode ist eine Erweiterung von FGSM, bei der FGSM mehrfach nacheinander angewendet wird.

$$X_0^{adv} = X, X_N^{adv} = Clip_{x,\epsilon}\{X_N^{adv} + \alpha sign(\nabla_X J(X_N^{adv}, y_{true}))\} \tag{3.2}$$

Der Wert α beschreibt hierbei die Größe der Änderung der Pixelwerte in jedem Schritt.

3.2.3 Methode zum Erreichen einer bestimmten Klasse

Die beiden vorhergehenden Methoden haben lediglich als Ziel bei dem entsprechenden Netz eine Fehlklassifizierung hervorzurufen. Um die als output eine bestimmte Klasse zu bekommen, wird die Iterative Methode leicht abgewandelt:

$$X_0^{adv} = X, X_{N+1}^{adv} = Clip_{X,\epsilon}\{X_N^{adv} - \alpha sign(\nabla_X J(X_N^{adv}, y_W))\} \tag{3.3}$$

Wobei y_W dem Wert der gewünschten Klasse entspricht.

3.3 Angriffe gegen eine Blackbox

Die dargestellten Methoden, Störbilder zu generieren haben alle auf dem ersten Blick einen gemeinsamen Schwachpunkt: Man benötigt Zugang zum Modell des Neuronalen Netzes, über den man bei industriellen Anwendungen als Außenstehender nicht ohne weiteres verfügen dürfte. Allerdings täuscht dieser erste Eindruck. 2016 zeigten Papernot et al. dass Störbilder, die für eine bestimmte Machine Learning Lösung generiert wurden, ebenso auf andere Lösungen anwendbar sind, solang diese Algorithmen die gleiche Aufgabe lösen. Das bedeutet, dass Störbilder, die zum Beispiel für ein Neuronales Netz zur Identifizierung von Straßenverkehrsschildern generiert wurden, für ein anderes, unbekanntes Netz und sogar für andere Strukturen, wie Logistische Regression oder Entscheidungsbäume verwendet werden kann. Um nun Störbilder für eine Blackbox zu generieren, ist es ausreichend, ein eigenes Neuronales Netz zu trainieren, das die gleiche Aufgabe löst und mit Hilfe der oben genannten Methoden Störbilder für dieses Netz zu generieren.

4. Methoden

Description of your used methods, algorithms and the actual work you did. This is often the longest and most detailed chapter...

5. Evaluation

Description of your evaluation and its results. Interpretation of these results should go in the next chapter.

6. Diskussion

Discussion/Conclusion section. Some questions to think about while writing this section:

- What is the outcome of your evaluation?
- Did it meet the required goals defined in chapter 2? Why/Why not?
- How can future work improve on your work?

Abbildungsverzeichnis

List of Algorithms

Literaturverzeichnis

[1] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.