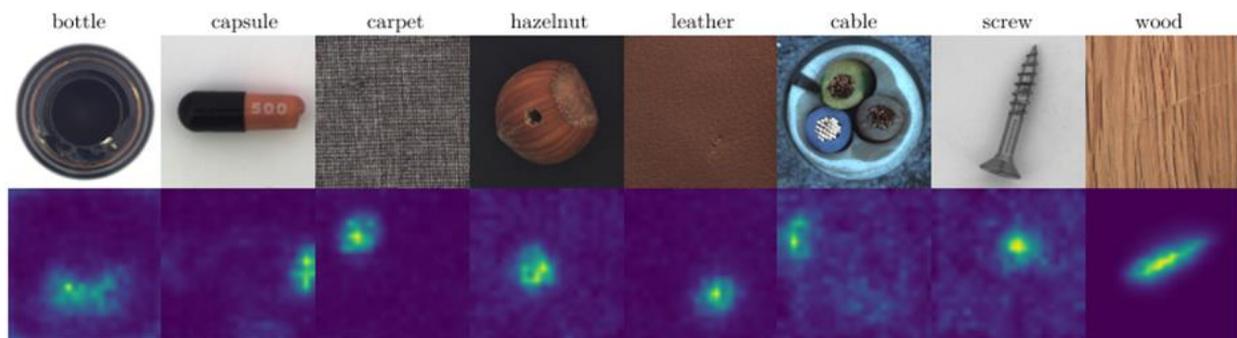


Masterarbeit

Transfer Learning für die visuelle Qualitätsinspektion auf einer Normalizing- Flow-Architektur



Ausgangslage

In der industriellen Fertigungstechnik werden häufig Assistenzsysteme für die visuelle Qualitätsinspektion von produzierten Artikeln eingesetzt. Ein Anwendungsszenario ist es auf Grundlage von Aufnahmen defektfreier Artikel ein datengetriebenes Modell zu trainieren, dass in der Lage ist die Aufnahmen weiterer anderer Aufnahmen desselben Artikels als tendenziell defekt oder tendenziell defektfrei einzustufen.

Problemstellung

Ein Modell, dass auf einem Artikeltyp trainiert wurde, lässt sich üblicherweise jedoch nicht unmittelbar auf einen anderen Artikel mit anderen Features übertragen. Das Trainieren weiterer Modelle für jeden Artikeltyp stellt sich als zu rechenintensiv und daher schlecht skalierbar heraus. Methoden des Transfer Learning können Abhilfe schaffen, indem ein Modell durch nachtrainieren auf neue Artikeltypen angepasst wird.

Vorgehensweise und Erwartete Ergebnisse

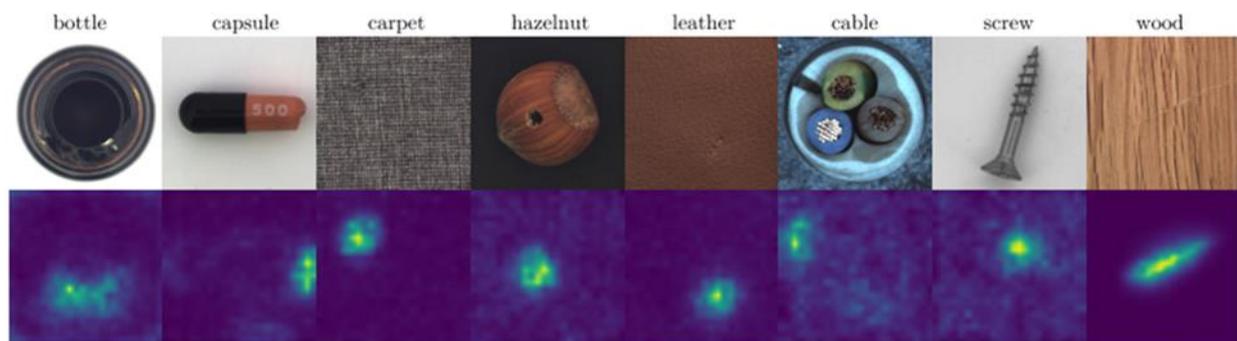
Ein Beispiel für eine Modellarchitektur, mit der eine Anomalieerkennung realisiert werden kann, nennt sich CS-Flow und macht sich Normalizing Flows (NF) zunutze. Auf dem öffentlichen Benchmarkdatensatz MVTec erreicht dieses Modell gute Ergebnisse hinsichtlich Anomalieerkennung. Im Rahmen der Arbeit soll untersucht werden, wie Methoden des Transfer Learning dazu genutzt werden können eine Übertragung von einer Fehlerbildklasse auf andere Fehlerbildklasse zu realisieren. Dazu muss der Inhalt des Datensatzes analysiert und eine Auswahl für eine Durchführung von Untersuchung hinsichtlich Transfer Learning passender Fehlerbildtypen ausgewählt werden. Die Umsetzung der Experimente erfolgt in PyTorch und die Evaluation der Transferierbarkeit der Qualitätsinspektion von einem Fehlertyp auf einen oder mehrere soll untersucht werden. Gute Programmierkenntnisse mit Python werden vorausgesetzt und Vorkenntnisse im Umgang mit numpy, pandas und PyTorch werden stark empfohlen.

Contact Person

Robert Maack | Tel.: +49 202 439 1714 | E-Mail: robert.maack@uni-wuppertal.de

Master's Thesis

Transfer Learning for Visual Quality Inspection based on a Normalizing-Flow- Architektur



Initial Situation

In industrial manufacturing, visual inspection of products is often supported with assistance systems. Using semi-supervised learning, a model is trained on pictures of a defect free product which enables the model to differentiate between pictures unseen non-defective and defective articles of the same product type.

Problem Definition

However, a model trained on pictures of products with a particular defect type is not guaranteed to be able to detect anomalies on pictures of products with another defect type. The training of one distinct model for each defect type is possible, but requires additional computation time and tends to not scale well. One possible solution to that issue is transfer learning with which a model is conservatively retrained on pictures of products with different defect type.

Methods and Expected Results

The architecture CS-Flow, which is based on Normalizing Flows (NF), is one example of an approach for anomaly detection in visual quality inspection. On the public dataset MVTec, this models in comparison to other models performs particularly well. In this thesis, transfer learning should be applied on this architecture. The student has to familiarize with the theoretical grounding, the implementation of CS-Flow and conduct research common methods of transfer learning. The content of the dataset has to be analyzed such that pictures with defect types that are suitable for conducting experiments for transfer learning, are extracted. Experiments have to be implemented in PyTorch and the evaluation of transferability from one to another defect type has to be demonstrated. Good programming skills in python required and prior knowledge in using numpy, pandas and PyTorch is strongly recommended.

Contact Person

Robert Maack | Tel.: +49 202 439 1714 | E-Mail: robert.maack@uni-wuppertal.de