

Personalisierung der Beiträge

Sie müssen bei der Abgabe genau angeben wer welchen Beitrag zu dem Projekt geleistet hat. Jede Leistungserbringung im Studium muss personalisiert sein!

— Prof. Dr. Bastian Beggel

Peter Stein

backend

- Erstellung und Paketierung des Python Packages `backend`
- Implementierung des `backend` in Django
- Erstellen der benötigten Datenbank models
- Ermöglichen der Speicherung von Bildern in der Datenbank
- Implementierung der REST Endpoints mit dem Django REST framework
 - Endpoint für das Auflisten der verfügbaren Kolorierungsmethoden
 - Endpoint für das Kolorieren von Bildern
 - Endpoints für das Auflisten und Löschen von bereits kolorierten Bildern
- Erstellen von Tests für das `backend` Package
- Konfiguration der Continuous Integration Umgebung für das `backend`
- Erstellen der Dokumentation (`README.md`) für `backend`
- Containerisierung (Docker) des `backend` packages

deepcolor

- Erstellen und Paketierung des Python Packages `deepcolor`
- Erstellen von Tests für das `deepcolor` Package
- Konfiguration der Continuous Integration Umgebung für das `deepcolor` package
- Erstellen des Command Line Tools für das `deepcolor` package
- Erstellung des Kolorierungsinterface `colorize_image` nach dem Strategy-Pattern
- Studieren des Papers und des Quellcodes des Projekts *Colorful image colorization* von Richard Zhang
- Implementierung des Kolorierungsnetzwerks *Colorful image colorization* von Richard Zhang
- Erstellen von Funktionen zum Automatischen Download der Deep Learning Models
- Erstellen von Funktionen zur Manipulation von Bildern
 - Konvertierung in verschiedene Farbräume
 - Konvertierung in Graustufen
 - Konvertierung zwischen numpy arrays und Pillow images
- Mergen und anpassen der Netzwerke `colormnet` und `zeruniverse`
- Erstellen einer Funktion zur Rückgabe der zur Verfügung stehenden Kolorierungsstrategien
- Erstellen einer Side-by-Side View für den Vergleich von verschiedenen Kolorierungsstrategien

- Erstellen der Dokumentation (`README.md`) für `deepcolor`
- Containerisierung (Docker) des `deepcolor` Packages

Sonstiges

- Studieren der Softwarelizenzen der verwendeten Projekte
- Erstellung der Projektlizenz (GPLv3)
- Zahlreiche Bugfixes
- Zahlreiche Refaktorisierungen
- Ständige Kommunikation mit den anderen Teammitgliedern

Sebastian Dauenhauer

Docker

- Erstellen der Container Images für `frontend`
- Erstellen der Build Pipeline für Docker
- Erstellen eines Reverse Proxy Webservers zur Verbindung von `frontend` und `deepcolor`
- Erstellen der Docker-Compose
- Erstellen eines Docker Scripts für GPU basiertes CNN Netzwerk auf Dockerbasis

frontend

- Erstellen der Frontend Schnittstelle zur Kommunikation mit dem backend mittels REST schnittstelle
- Erstellen eines ersten Design Prototypen
- Erstellen von React Komponenten zur modularisierung des Frontends
- Neuer Design Entwurf des Frontends basierend auf [HTML5Up](#) Design
- Anpassen des Designs auf mobile Geräte (responsive)
- Erstellen von TypeScript Komponenten
- Erstellen der Build Pipeline für `frontend`
- Erstellen der Dokumentation (`README.md`) für `frontend`

deepcolor

- Erstellen des Colornet Models anhand des Papers `Let there be Color` und [einer PyTorch implementierung](#)
- Einbinden von `colornet` in Kolorierungsinterface

Training

- Erstellen eines Scripts zum Download sowie Verarbeiten der Trainingsdaten (places205)
- Erstellen eines Trainingsscripts anhand der im Paper beschriebenen Parameter
- Versuch Training durch andere Parameter und Algorithmen zu verbessern
- Studieren von Verschiedenen Optimizern und Trainingsmethoden
- Training von Colornet auf mehreren GPUs von Skynet
- Erstellen der Dokumentation (`README.md`) für `Training`
 - `deepcolor/README.md` -> Abschnitt: Training
 - `train/README.md` -> komplett

Sonstiges

- Zahlreiche Bugfixes
- Zahlreiche Refaktorisierungen
- Ständige Kommunikation mit den anderen Teammitgliedern