Hochschule
München
University of
Applied Sciences

Fakultät 7

Prof. Dr. Markus Friedrich

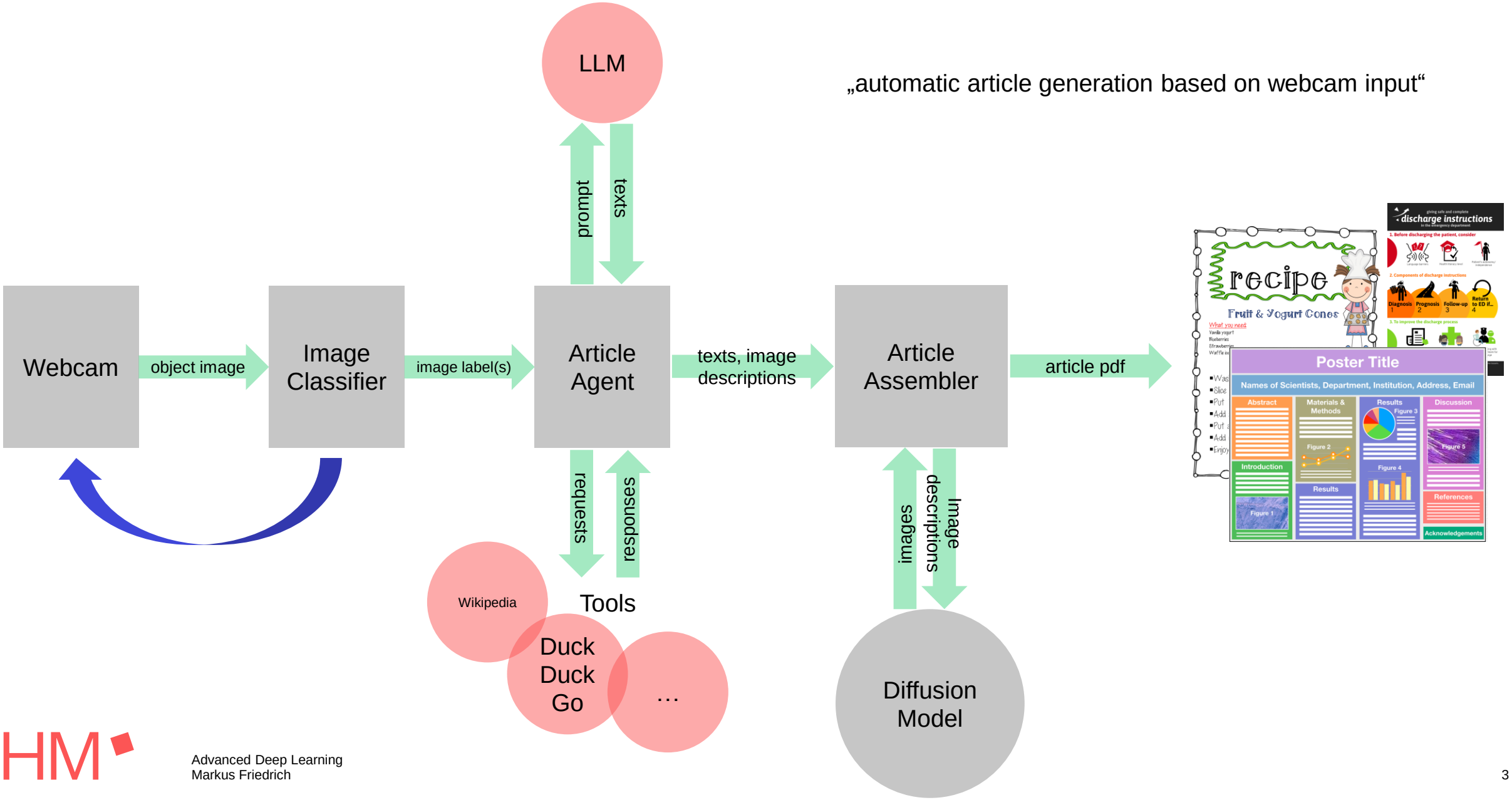
Advanced Deep Learning
Team Project

Goals

- Work together in a team on a machine learning task that is solved with several deep learning techniques.
- Implement DL models and use existing implementations.
- Learn how to evaluate DL models.
- Use state-of-the-art tools.

System Description

„automatic article generation based on webcam input“



System Description - Modules

Webcam

- Records images when triggered by user input.
- Tools / packages: opencv-python

System Description - Modules

Image Classifier

- Receives, pre-processes and classifies webcam images
- packages: pytorch, pytorch lightning, captum, autodistill, (optuna) | services: w&b

	Implementation	**Training**	**Evaluation**
AlexNet	implement	from scratch • on ds1 • on ds2	On ds1 and ds2: • Test set results (sensible metrics) • Training loop profiler results (pytorch lightning) • Loss curves • Worst and best case examples with explainable visualization (captum)
Reset-50	use existing	pre-trained on ImageNet • fine-tuned on ds1 • fine-tuned on ds2	
Vision Transformer	use existing	pre-trained on ImageNet • fine-tuned on ds1 • fine-tuned on ds2	

- ds1: existing data set (e.g. found on kaggle)
  ds2: extended dataset with additional self-curated data. Data auto-labeled (autodistill).
  Selection of images based on evaluation and examination of best and worst case examples (captum).

System Description - Modules

Image Classifier

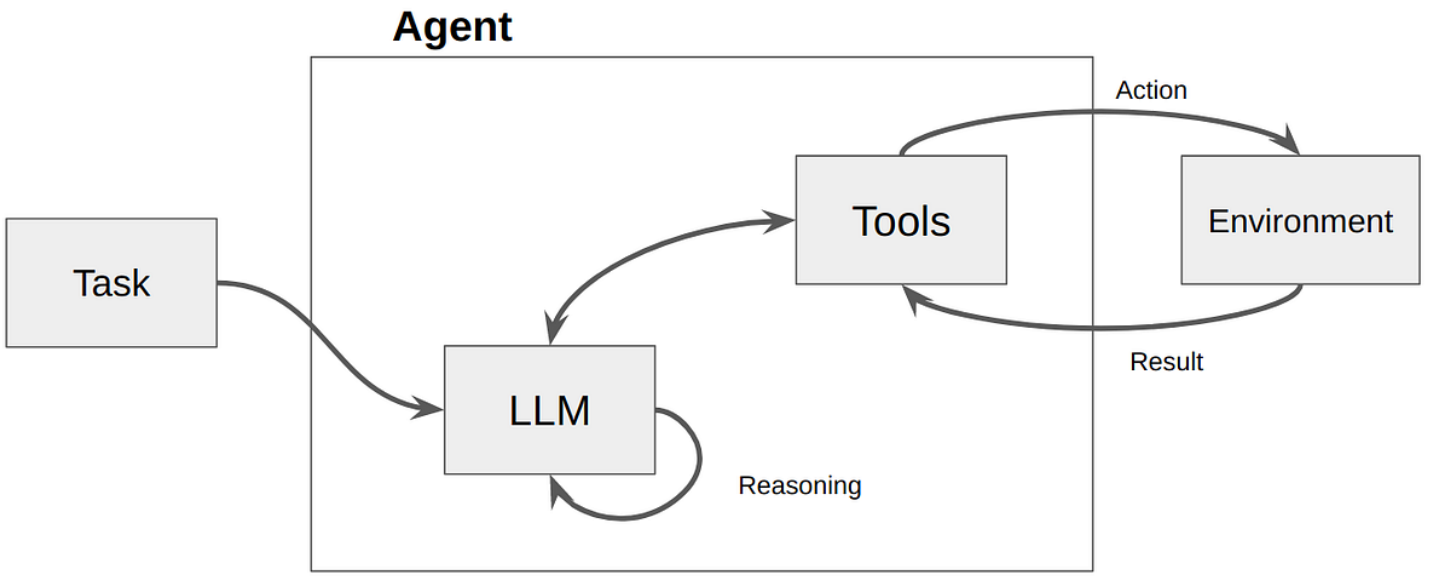
- Training:
  - Choose sensible train/validate/test split
  - Experiment with hyper parameters!
  - Experiment with data augmentation!
  - Store run results on w&b!

- Bonus task: Tune hyper parameters automatically (optuna)

System Description - Modules

Article Agent

- The article agent uses LLM agents (langchain) to interact with external services.
- These external services (e.g. wikipedia) are used to gather information based on the set of predicted image class labels.



Source: https://miro.medium.com/v2/resize:fit:1200/1*5TnpUZnp4-sq8TuJGYe_-w.png

System Description - Modules

Article Assembler

- Takes texts and image descriptions from the article agent and images from the diffusion model and generates an article

- An Article contains:
  - Text paragraphs (min. 1000 words, min. 4 paragraphs)
  - Figures with caption (min. 4 figures)

- Proposed transformation pipeline: take input and put it in a markdown template, use pandoc for markdown → PDF conversion

- **Important:** text must make sense (facts can be LLM-ish off).

System Description

Overall System & Usability

- The system connects the aforementioned parts seamlessly in a single application.
- The UI can be very restricted (command line is also fine).
- User guidance:
  1. The user is prompted to point the webcam at an object and press a key / button.
  2. The system responds with an “image recorded” message and goes to 1).
  3. If the user presses a certain start-key / button, the generation process starts with the set of recorded images.
  4. The system returns the article as a PDF.

System Description

First steps

1. Form teams
2. Come up with a good domain
   (potentially sensible articles? suitable data sets available? Search e.g. on kaggle / papers with code, google dataset search)
3. Setup infrastructure (git, service accounts)
4. Distribute work
   (**important:** equal workload)