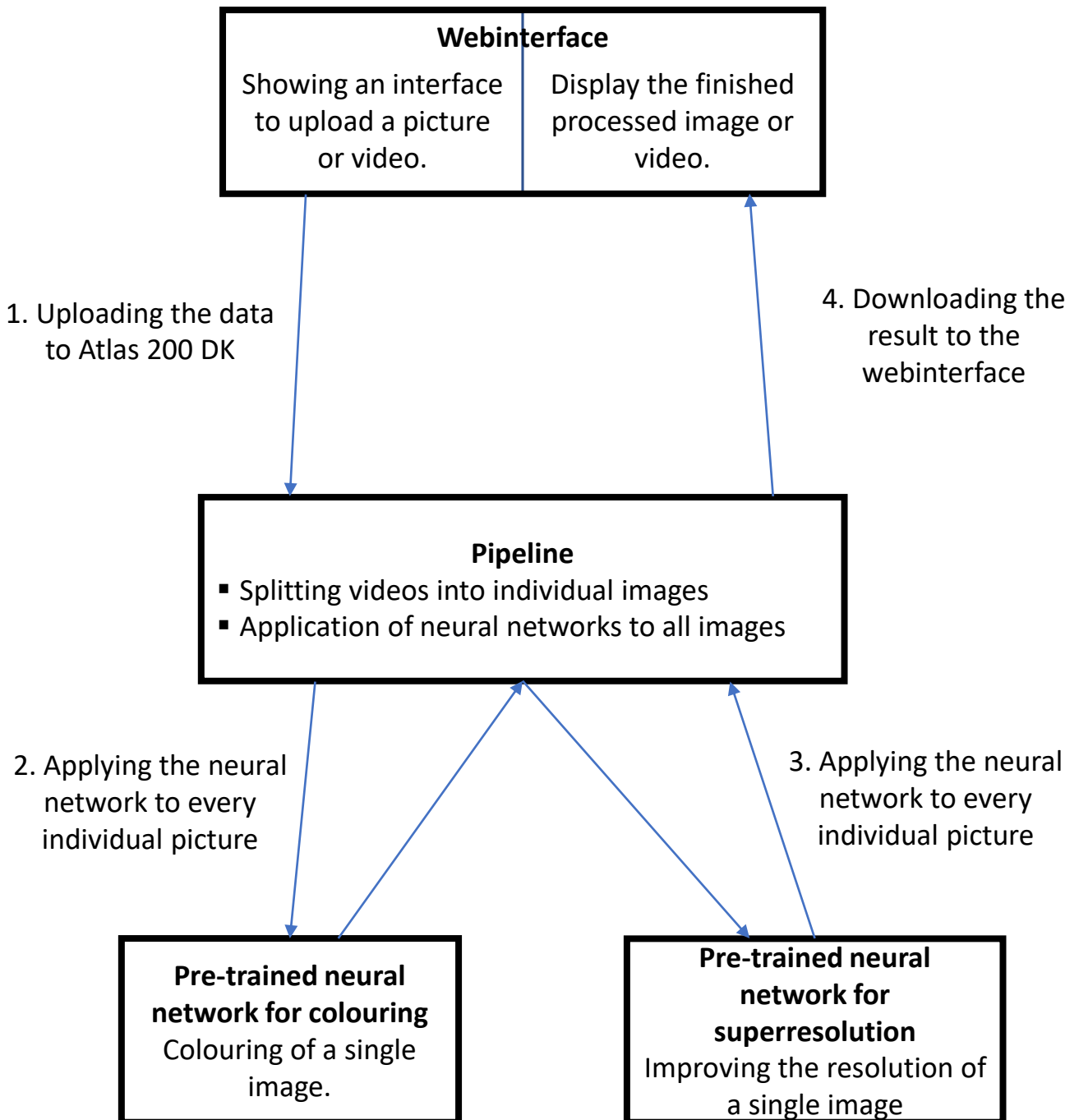


# Summary of runtime components



# Summary of code components

## Webinterface frontend

Building a frontend with html, css and js

- tailwindcss
- dropzonejs
- jquery

## Webinterface backend

- Building a backend with python-based flask
- Communication to the frontend with json

## Pipeline

Building a python pipeline with the functionalities:

- Splitting videos into single pictures
- Applying the neural networks to the picture
- Reassembling videos

**Pre-trained neural  
network for colouring**  
Single datafile

**Pre-trained neural  
network for  
superresolution**  
Single datafile

# Summary of the underlying technology stack

/	Context	Name	Version	License
1	Webservice backend	flask	1.1.2	BSD-3-clause (BSD-new)
2	Webservice frontend	tailwindcss	2.1.2	MIT
3	Webservice frontend	dropzonejs	5.9.2	MIT
4	Webservice frontend	jquery	3.4.0	MIT
5	OS	Ubuntu	18.04	GPL
6	Developer board	Atlas 200 DK	1.73.0.0	
7	Pipeline backend	Python	3.7.5	GPL
8	Toolkit package	Ascend Toolkit	20.0.RC1-arm64-linux_gcc7.3.0	GPL
9	Toolkit package	Ascend Toolkit	20.0.RC1-x86_64-linux_gcc7.3.0	GPL
10	Toolchain platform	Mind Studio	2.3.3	

# Textual explanation of the diagrams and choices

## Runtime components

The first element is the web interface. Its task is to accept the user's input and transfer it to the AMOS board on which the pipeline runs.

This pipeline accepts the file and checks whether it is a video or an image. An image is forwarded, while a video is first broken down into individual images. All other file formats are rejected with an error message.

The pipeline applies all neural networks to each image one after the other. If it was a video at the beginning, it is now reassembled.

The result is sent back to the web interface and displayed to the user.

## Code components and technology stack

The web interface consists of a frontend and a backend. The frontend represents the interface for the user, while the backend takes over the communication with the Atlas 200 DK.

The frontend is designed in html, css and javascript and built with the frameworks tailwindcss, dropzonejs and jquery. The reason for this is that the frameworks are excellently suited for the tasks and the team already has experience with the frameworks.

The backend is based on python, more precisely the framework flask. The reason for this is that python is the best-known programming language in the project team and also contains features that are needed for communication with the Atlas board.

The pipeline is also built-in python because the programming language was specified by the client and the communication with the pre-trained neural networks works so well.

The pre-trained neural networks for colouring and superresolution are available as a file and are called up and used by the pipeline. These are supplied and specified by the client.