NTNU

Bachelor Thesis
GUI 4 deep-doLCE

Authors

Runar Astvaldur Hedin

Aleksander Kokowski

Qeis Khaled Qeis

Bachelor of Science in Engineering - Computer Science

BIDAT

Department of Computer Science

20.05.2022

Supervisors Sony George

Steven Yves Le Moan

SAMMENDRAG

Tittel:	GUI 4 deep-doLCE	Dato: 20.05.2022
Deltaker(e):	Runar Astvaldur Hedin Aleksander Kokowski Qeis Khaled Qeis	
Veileder(e):	Sony George Steven Yves Le Moan	
Evt. oppdragsgiver:	Dr. Giorgio Trumpy	
Stikkord/nøkkelord (3-5 stk):	GUI, Linseformede filmer, Dyp læring, fargerekonstruksjon.	
Antall sider: 41	Antall vedlegg: 3	Publiseringsavtale inngått: Åpen

Sammendrag:

Hensikten med dette prosjekter er å utvikle GUI for en dyp lærings film og bilde fargerekonstruksjon programvare kalt deep-doLCE. Vår programvare må være forståelig og brukbare for alle med som har skannet linseformede filmer, uansett filmarkiverings kunnskap. Vi undersøker en enkel og intuitiv GUI som leder brukeren til neste knapp ved å skjule og vise knappene brukeren skal trykke på.

ABSTRACT

Title:	GUI 4 deep-doLCE	Date: 20.05.2022
Participants:	Runar Astvaldur Hedin	
	Aleksander Kokowski	
	Qeis Khaled Qeis	
Supervisor(s):	Sony George	
	Steven Yves Le Moan	
Employer:	Dr. Giorgio Trumpy	
Keywords: (3-5)	GUI, Lenticular films, Deep learning, Colour reconstruction.	
Number of pages: 41	Number of appendixes: 3	Availability: Open

Abstract: The purpose of the project is to develop a Graphical User Interface for a deep learning film and image color reconstruction software called deep-doLCE. The application needs to be understandable and usable by anyone with a scanned lenticular film, regardless of any film archiving knowledge or not. We propose a simple and intuitive GUI that guides the user to their next appropriate action by hiding and showing the buttons they should press next.

1 Contents

List of Figures

Preface

This thesis was written by a team of three computer science students at the Norwegian University of Science and Technology. We chose this assignment because of our interest in deep learning. We were confident that our knowledge and skillset developed during our education all met the requirements of the task given by the client Giorgio Trumpy.

We would like to thank our client Dr. Giorgio Trumpy for allowing us to take on this interesting and challenging task. We would also like to thank him and his colleagues for all they taught us during the project.

We would also like to thank our supervisors Sony George and Steven Yves Le Moan for their valuable advice, feedback and insight during our work.

Terms and abbreviations

GUI – Graphical User Interface. Displays objects that convey information and represent actions that can be taken by the user.

Lenticules - any of the minute lenses on the base side of a film used in stereoscopic or color photography.

doLCE – A method that detects lenticules and colorizes that image.

ROI – Region-Of-Interest. Samples within a data set identified for a particular purpose. A chosen area within an image chosen to colorize.

RGB – Red Green Blue. The three hues of light used in imagine and by technology to create colours.

Miro - Online whiteboard for team collaboration. Miro's whiteboard app for tablet and mobile gives you the tools to collaborate with boards that put projects and context all in one place.

CPU - central processing unit. The electronic circuitry that executes instructions comprising a computer program.

GPU - graphics processing unit. A specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device.

IDE - An integrated development environment. Software for building applications that combines common developer tools into a single graphical user interface (GUI).

Frontend – What the user interacts with for a more user-friendly and simple experience. Also referred to as front end or front-end.

Backend – What happens behind the scenes without the user seeing it. Also referred to as back end or back-end.

OS – Operating system. system software that manages computer hardware, software resources, and provides common services for computer programs.

Framework - Resources and tools used to build and manage applications.

NumPy – A Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

8-bit image – An 8-bit image has a 2^8, or 256 levels of colours and tones that can be manipulated.

16-bit image – A 16-bit image has 2^{16}, or 65.536 levels of colours and tones, resulting in 256 times more levels and a far greater detail of the image.

2 Introduction and relevance

Lenticular films were some of the earliest technologies that permitted to capture full colour information in motion. These films however were only used between the 1920s and 1940s. This encoded colour information can be processed today to give the greyscale images colour. The deep-doLCE project proposes the first accurate digitization and colour reconstruction process for lenticular films that is robust to artifacts. Thus, the goal for our project is to create a GUI for deep-doLCE. The client wanted the GUI to be clean and simple with the hopes that anyone can use it and hopefully this should lead to archiving more of these rare historical films.

The client stated that the GUI will connect to deep-doLCE's backend and should have four main buttons that are intuitive to use. Some of these buttons required additional work in order to be compatible with the backend, such as the standard 16-bit films into 8-bits because deep-doLCE only accepts 8-bit films. Another part was to improve the speed of colour reconstruction by making the software run on the GPU rather than the CPU. Images of the first concept, early and final versions of the GUI can be found later in the thesis.

2.1 Tasks

The goal of the thesis is to design and test a graphical user interface for deep-doLCE. The GUI must allow the user to upload the input image-sequence or video file (handling different file formats), select a region of interest, launch the colour reconstruction process, and monitor its progress. We must also not edit the backend source code provided by the client in any way (as the client requested).

2.2 Our contribution to the task

Since deep-doLCE does not have a user interface it limits its usability to programmers, excluding most people from using it. In addition to that, deep-doLCE accepts only one image at a time with a specific format (8bit grayscale PNG images), which means the software does not process videos and limits the file formats accepted by software and image sequels should be processed one at a time.

Our task is to create a GUI using python (as stated by the client because that’s the language deep-doLCE was developed in) that allows the users with no basic programming skills whatsoever to use deep-doLCE easily and select any uncompressed lenticular image or video format they want, in addition to the possibility of selecting multiple images or image sequences and the opportunity to select the region of interest.

In addition to our main task, we decided to create a test button that saves the user's time by selecting random frames out of the image sequence after colorization and letting the user check and make sure if it was colorized correctly. This can save a lot of time because we run a single frame and check results, rather than all of the hundreds or thousands frames before seeing the results. Lastly we allow the user to select different types of output format.

2.3 Limitations

Speed of colour reconstruction is the biggest limitation we faced because deep-doLCE has a runtime of 3 to 5 minutes per frame. When programming or editing the program, we must test that all aspects of the program are running successfully and that takes a large amount of time due to the process time of the deep-doLCE code provided to us by the client. A side limitation was also our own computers since the colorization speed relies on computational power.

2.4 Earlier experience

Through our studies, we have acquired knowledge about the whole development process, in particular front-end programming and image processing. We also got experience in project planning. This is all something we know will be relevant and useful for this project. Besides this, the group has also had previous experience with working in teams during a diversity of projects during our studies. We have learned a lot about the tools necessary for working with such a project and were both excited and confident to use our knowledge with this real-life task.

2.5 What we had to learn

Although we all have previous relevant experience through our studies, there was still something to be learned before, and during, the project. Specifically, the group lacked important knowledge about the GUI tools in python. Our learning experience are mentioned throughout the whole report.

We also had to learn about lenticular films and films archiving in general so we could understand what and why we had to include some features such as flipping an image.

2.6 Thesis structure

The thesis is structured into the following 10 chapters:

1 Introduction The reader is introduced to the group and the project.

2 Project fundamentals In-depth details on lenticular films and deep-doLCE, and why we chose this project.

3 Development Process When and how the various parts of the project were developed and what development model we followed to reach the goal of the project.

4 Technologies Here the reader can get more information about what tools were used by the team to solve the various problems and tasks that we faced.

5 Design and implementation The explanation for certain design choices, why, what, and how things were done in a certain way to the make the application as easy as possible to use, while also enhancing the user experience.

6 Choice of the development environments The developer tools we used to develop the software.

7 Code quality Summary of our standards for the quality of the code, which helped us understand each other's code, as well as hopefully all future developers after our work has been turned in.

8 Testing The different types of testing we did.

9 Discussion The result of the work we did, and discussion based on the feedback from the usability tests.

10 Conclusion Here the reader can get more details about the group's thoughts and our experience at the end of the project, as well as a summary of the entire project. The results were discussed by the group.

3 Project fundamentals

This chapter will go into more detail of what the software does, how it does it and why we were interested in taking on this challenge.

3.1 Background

In order to understand how our software works we must first understand what lenticular films and deep-doLCE are. This sub-chapter focuses on informing the reader about lenticular films and deep-doLCE.

3.1.1 Lenticular films

Lenticular films are rare films that were only shot between 1920s and 1940s that contain encoded colour data. Lenticular films work just like any film or image where it takes in lights rays that get redirected to a single point to create a sharp image. Where lenticular films are different lies in the lens. Most images are taken with stacked concave and convex lenses that work similar to an eye, lenticular films instead characterized by a vertical array of hundreds of cylindrical lenses. According to deep-doLCE: A New Machine Learning Approach for the Color Reconstruction of Digitized Lenticular Film: “consisted of a black-and-white film embossed with vertical cylinders, called lenticules, and an RGB filter placed in front of the camera lens. This combination allowed to encode the color information as silver densities in the horizontal spatial dimension within each lenticule. After being exposed in the camera, the film underwent reversal processing. Its visualization was done with a dedicated projection device that reversed the recording process, using an RGB filter in front of the lens to create color images from the black-and-white film.”[1]. Comparison images in figures 2-1 and 2-2.

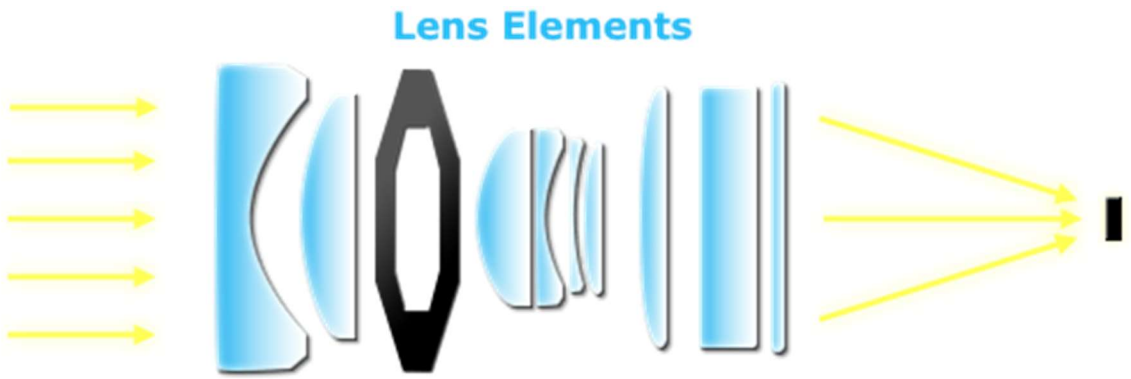


Figure 3-1 Most common camera lenses [2]

[1] deep-doLCE: A New Machine Learning Approach for the Color Reconstruction of Digitized Lenticular Film

[2] *UNDERSTANDING CAMERA LENSES*

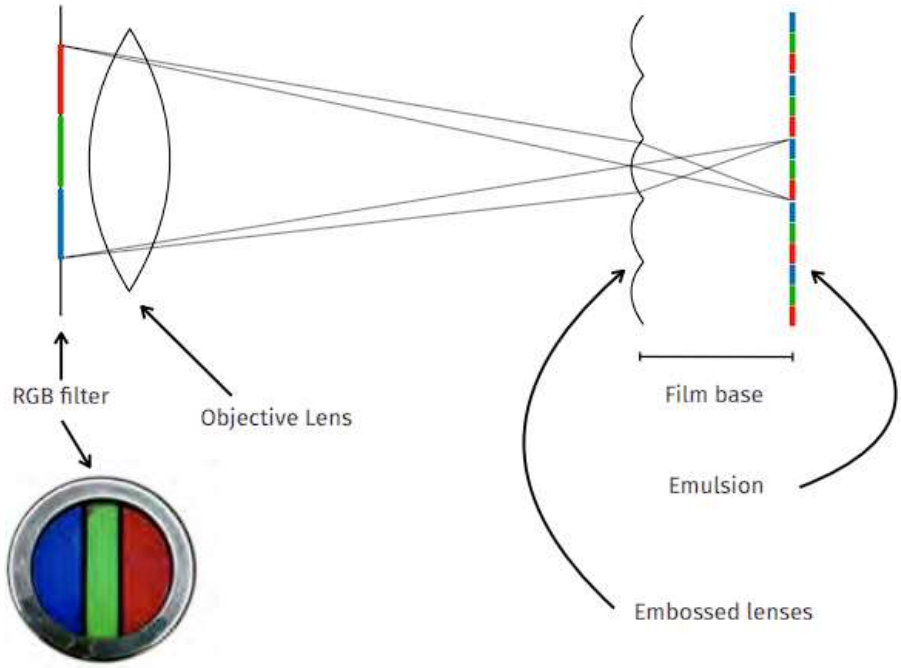


Figure 3-2 Lenticular film lens [3]

As the reader can see on figure 2-2 the lenticular film will use the RGB filter and the tiny cylinders to focus the light colour information on different parts of the emulsion, resulting in the colour information being encoded on the image. This will result in thin vertical streaks in the film which, when overlayed with the RGB filter to process the image, will allow a special projection device to reconstruct the colours. Example of the streak and filter below in image 2-3.

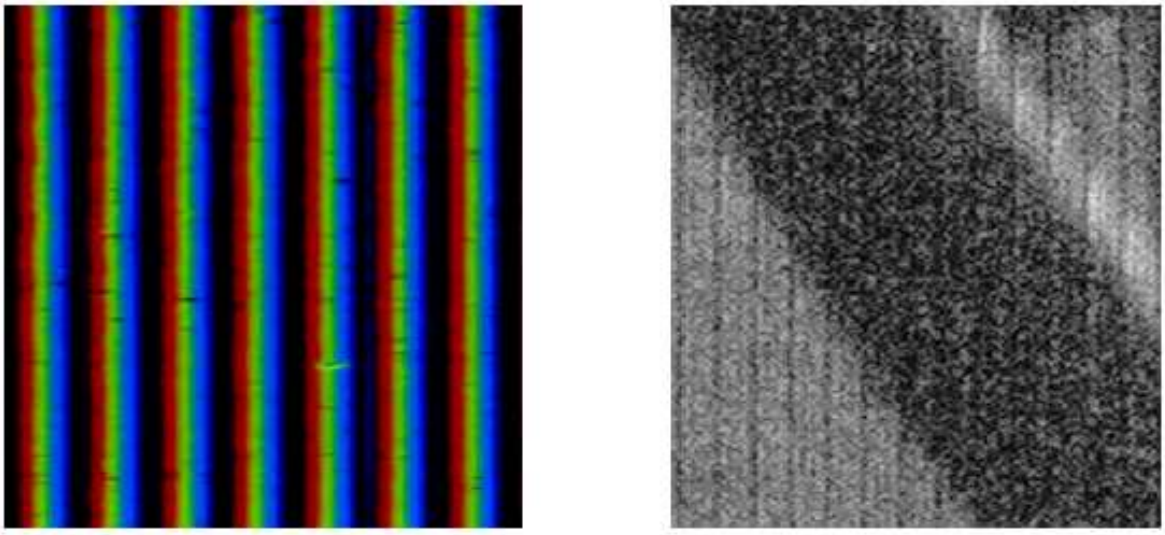


Figure 3-3 RGB filter on the left and detail of a diagonal edge on a scanned lenticular film to the right [3]

As of 2022, lenticular lenses remain used to this day, through 3D effects in images and by making images appear to move if seen from different angles, or even combining two images where you see

[3] A Deep Learning Approach for Digital Color Reconstruction of Lenticular Films

different ones depending on the angle. These lenses also had a short life in 3D television as of 2010, with only a few continuing to manufacture them.

3.1.2 Deep-doLCE

Reconstructing the rare lenticular films requires a special, obsolete projection device and is a time-consuming process to boot. Deep-doLCE aims to bring this colorization into the modern world via deep learning, potentially both removing the projection device requirement as well as speeding up the process, and possibly enhancing the results.

“Some of the earliest home videos were shot on 16mm lenticular film between the 1920s and 1940s. This incredibly special film is embossed with a vertical array of hundreds of tiny cylindrical lenses used to capture scenes in colour on a black and white silver emulsion. The most efficient approach to obtaining digital colour images of these historic films is to scan the silver emulsion at high resolution and have software extract the encoded colour data. While most software works on the localization of the lenticular screen, this is the first and most complicated step in colour reconstruction. In some cases, a "classic" signal processing method has proven its worth and delivered successful results, but often adverse factors (damaged or warped film, scanning problems) make it difficult to get successful results.

Deep-doLCE is explores a more advanced and robust method that uses a large dataset of digitized lenticular films to train new deep learning software. The aim is to create an easy-to-use software that will revive awareness of lenticular colour processes and thus make these valuable historical colour films accessible to the public again and secure them for posterity.”

The reader can find an image of deep-doLCE’s results and process below, courtesy of our client Giorgio and his fellow colleagues’ research.

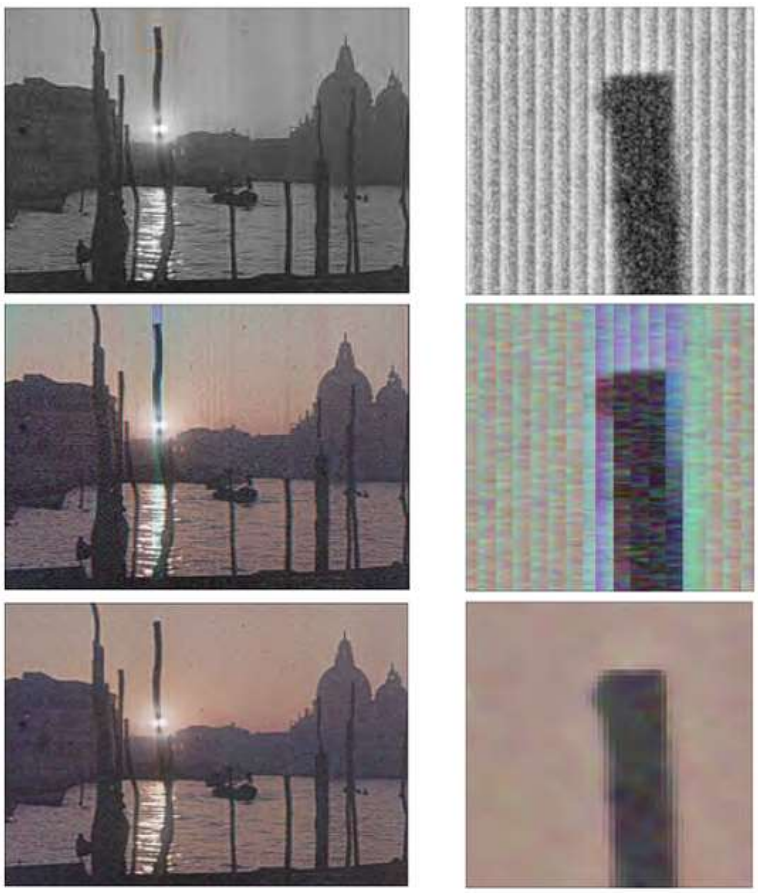


Figure 3-4 Example of a scanned lenticular film(top), doLCE (center) and deep-doLCE(bottom)[3]

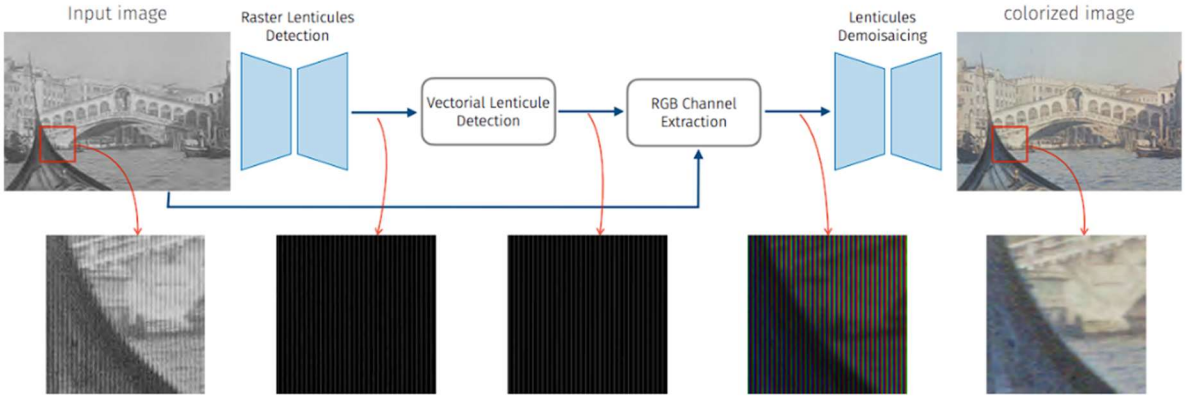


Figure 3-5 Overall pipeline of the proposed method.[4]

[4] A Deep Learning Approach for Digital Color Reconstruction of Lenticular Films

3.2 Detailed GUI objective and requirements

As mentioned in the introduction, our project is to create a GUI for deep-doLCE that is easily usable by not just experienced archivists but anyone that has a scanned lenticular film. This software must also run on both Windows and MacOS but has no requirement for other OS.

The GUI will need four buttons, Upload media file, Select ROI (which was renamed to Crop later into the project), Test, and Run. In addition to this we added the Help button to hopefully clear up any confusion.

The 'Upload media file' button opens your file explorer like any standard file upload button except it limits the file types to only the ones the deep-doLCE backend can process. We need to deconstruct a film file into individual frames so that we can run the code on that file. The button should also let you select one or more image to be processed. The image then also needs to be converted from 16-bits to 8-bits.

The 'Select ROI' button should let the user select a specific area or areas they would like colourized.

The 'Test' button should run a quick preview of the uploaded image or film and let the user confirm whether the flipped or un-flipped image has been colourized correctly. The reason why we need to display these flipped or un-flipped images is because of the RGB filter used in the lenticular film as mentioned in the previous chapter. A scanned image might already have been flipped or not which can result in the filter switching the colours from RGB to BGR which will result in a very off-colour image. If neither of the images are correct, then the user can choose to run another random frame from the image sequence. There are multiple reasons why an image can fail to colourize correctly, like smeared image, not actually a lenticular film, or the resolution is too low. However, the most common reason is that the image is flipped and so the RGB ends up being flipped. We circumvent this issue by always processing a frame twice, one regular frame and one flipped. The user then confirms which frame is colourized correctly which will then run the colour reconstruction function on every single image in the image sequence.

The 'Run' button starts the colour reconstruction of the image sequence and should allow you to save the resulting image sequence into the file type of your own choice.

An additional task if we have time is to improve the speed of colour reconstruction.

3.3 Why we chose this task

The group wanted an exciting and challenging project, that would give us the opportunity of developing experience that could later help us as an advantage when entering the job market. The group agreed that the deep-doLCE project fulfilled these wishes, so it was immediately put into a list of possible bachelor project. After reviewing the project description and discussing it, we acquired a big interest in it, and we thought that it would be a perfect project to have as our bachelor thesis. The deep-doLCE project is a front-end development project, which is something that we found very intriguing, and we thought that we could obtain a lot of learning experience from.

4 Development Process

4.1 System development model

Using system development models during development helps in successfully being able to develop projects that meet certain requirements and premises. It helps organize and structure work and allows you to easily track what tasks are being performed, as well as when and how the development process should be carried out. There weren't many constraints and wishes on how the development process should be carried out by the client for this project, but we still had some criteria that we had to adhere to.

4.2 Project characteristics

The characteristics of the project that were considered when choosing the development model are listed below.

1. Set a delivery deadline
2. Small development team of three
3. Frequent meetings with the client to receive continuous feedback on the development and get innovative ideas or room for improvement.
4. Requirements and functionality of the project is mostly figured out but there were still room for improvements and innovative ideas by both the group and client.

4.3 Choice of system development models

Initially, the customer had submitted some core requirements/functionality, but it was expected that further requirements for the project would arise as the initial requirements were not fully detailed. This meant that choosing an incremental development model was not a viable solution. An agile development model would instead allow us to continually receive suggestions & feedback and make fundamental changes to ongoing functionality. The group leaned strongly towards Scrum, a well-known practice we used in the past. Combined with Scrum, we adopted the Kanban practice of keeping track of backlog elements and their status on a Kanban board.

4.4 Process execution

The project was divided into eight sprints, each sprint has a time frame, to keep track of the process and each sprint has a specific task.

4.4.1 Purpose of each sprint

The eight sprints were distributed evenly over the period February 1. - April 20 as seen in Appendix A, figure 6.2. In the planning phase the group discussed and set up each sprint and decided to give each sprint a specific purpose and set an incredibly detailed plan for each one, to make sure the process is going smoothly, and the project is finished on time, due to one of our members needing a surgical operation then catching COVID-19 at the hospital we got a bit behind schedule but we managed to finish our project on time.

Sprint 1: Identify the most suitable tool and start designing the interface.

Sprint 2: Include the buttons that let navigate the file system and select input files and destination folder.

Sprint 3: Improve compatibility of deep-doLCE with the most common file formats used by film scanners.

Sprint 4: Offer to the user the possibility to select a Regio-Of-Interest (ROI).

Sprint 5: Let the user choose the file format for the output, including a down sample option.

Sprint 6: Run a test reconstruction on a single frame (a pre-flip left-right could be necessary).

Sprint 7: Create the RUN button that start the reconstruction of the whole movie.

Sprint 8: Give the user feedback on the status of the process (i.e., output image, ETC, etc.)

4.4.2 Scrum board

The Kanban board (Figure 3-1) has been a very important tool during the process of the project, and we used the Kanban-style list-making application Trello for this. The main reason for using Trello is to keep track of what needs to be done and to put a time limit for every task. That helps prevent us from falling behind.

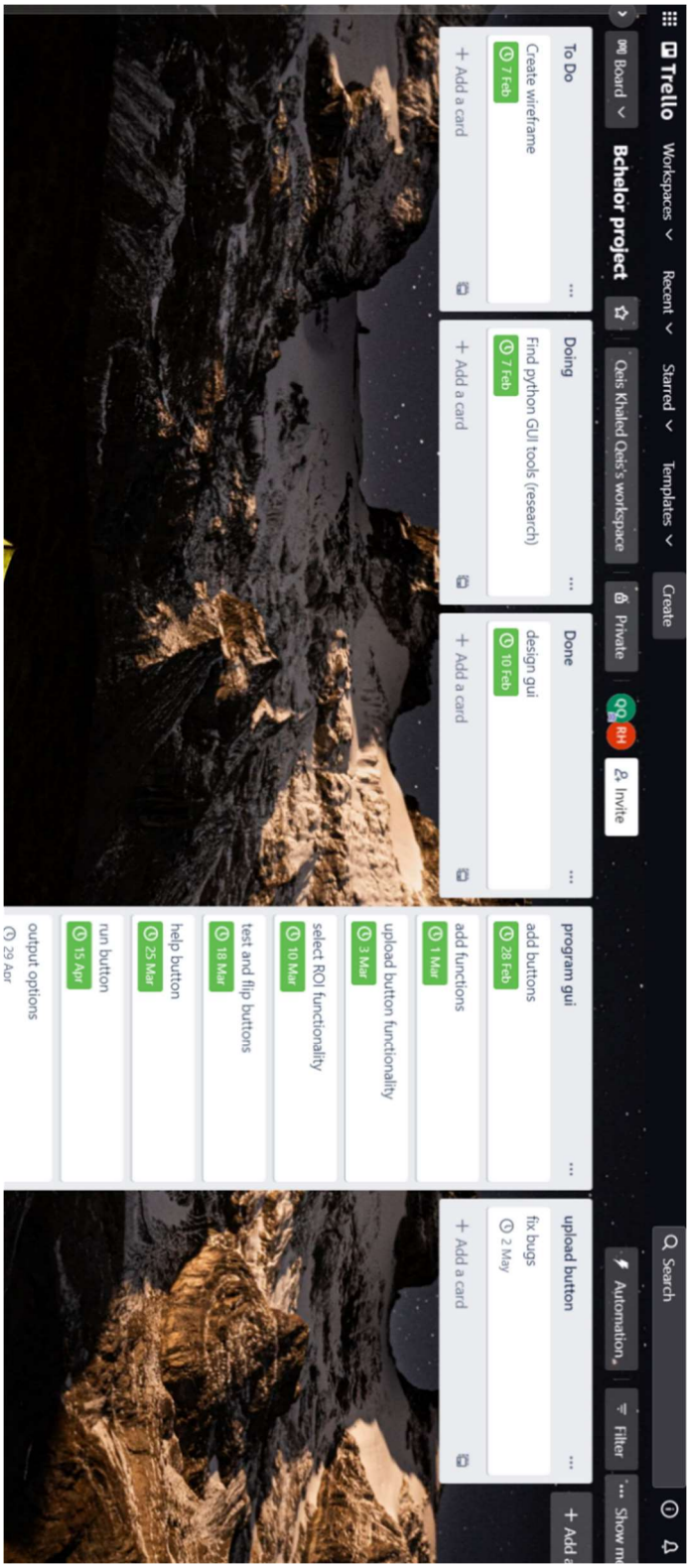


Figure 4-1 Scrum board in Trello

4.5 Project time usage

The successful implementation of a bachelor thesis project requires a good amount of time. In order to achieve our goals within the project, we decided as a group to set ourselves the goal of working on the project at least 30 hours a week or 6 hours a day. This amount was estimated as the necessary workload for a subject worth 30 study credits. The team used Toggle and Excel to track time spent, and tag it with different types of activities. The time usage per activity is shown in Figure 3-2.

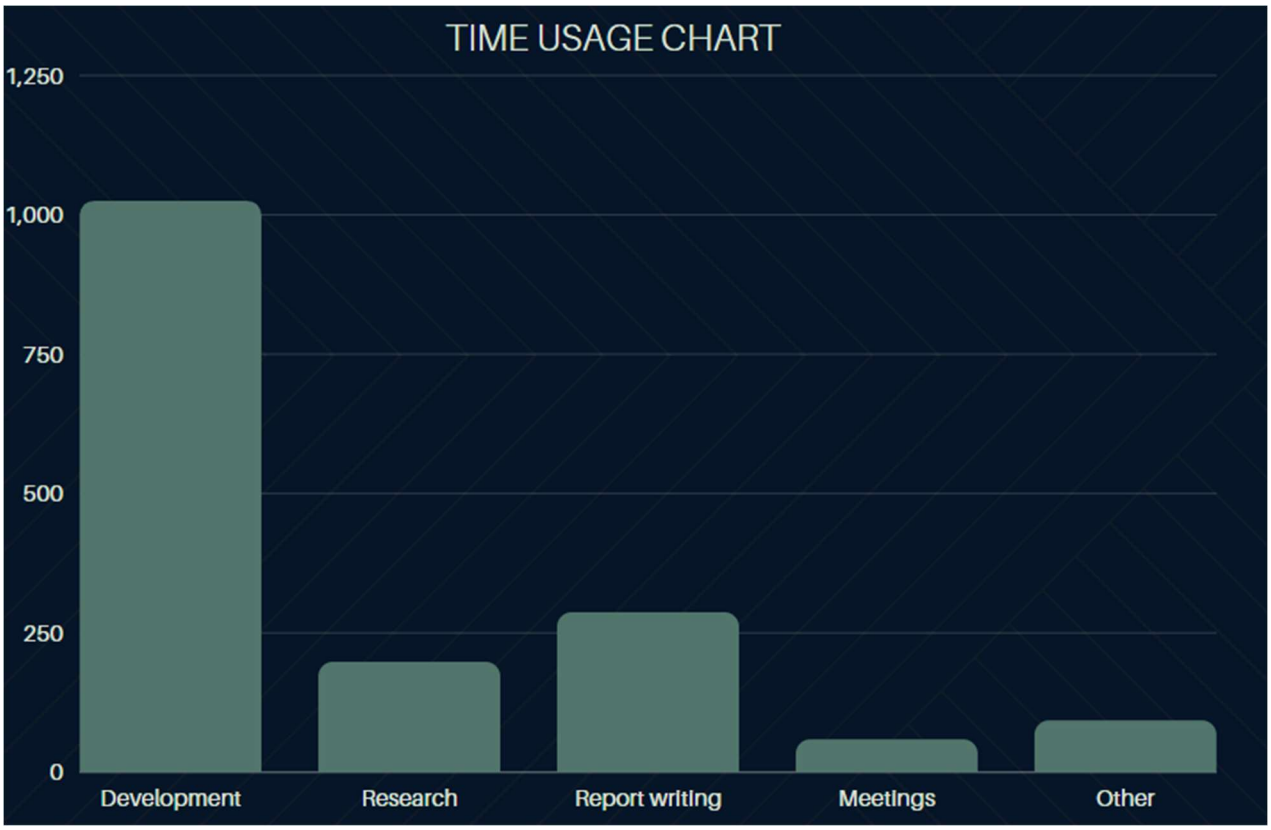


Figure 4-2 Working hours spent on different activities

5 Choice of technologies and methods

In this chapter we will go over the technologies and methods we used for our project.

5.1 Technology

We needed to use a python GUI framework to make the GUI since that is the language deep-doLCE was developed in. This was the only requirement in terms of tools made by the client since we only had to develop the frontend. In this chapter we will discuss which framework we chose, and why we chose it, among other useful tools.

5.1.1 Framework

After spending some time researching different GUI frameworks, we narrowed our choices down to Kivy or Tkinter. We chose the former initially, but one team member ran into issues during installation, which forced us to use Tkinter in the end instead. Tkinter was easy to set up and learn in addition to a lot of additional resources through Stack Overflow etc.

5.1.2 Excel

We used Microsoft's Excel to note down the start and end time of our meetings, as well as to write short summaries of each meeting. The longer, in detail notes were posted on our discord channel instead. Excel is a simple but powerful tool for this because it allows you to easily calculate total hours and minutes spent on meetings and lets you categorise in whatever way you choose.

5.1.3 Communication programs

Most of our communication and meetings were held online, so when we were not meeting physically, we had to use apps to communicate. We chose Discord for our internal group chat, Microsoft Teams for weekly meetings with the supervisors, and either Microsoft Teams or Zoom to talk to our client.

5.2 Methods

5.2.1 Wireframing

Considering that all our development is the GUI, it becomes vital to design a good, simple, yet user-friendly GUI and so having a basic wireframe idea is very important. Wireframing is a time-cheap option for creating mock versions of product. This helps us both recognize as many of the constraints as possible, and to understand the product itself. Together as a team and with the input of the client, we can figure out what works and what doesn't, ensuring the product is of higher quality, as well as ensuring that we're creating a product that the client is both happy with and wants.

The tool we chose for wireframing in this project is the free, interactive, and collaborative online whiteboard Miro. Not only does it offer real-time collaboration but also multiple free templates. [5]

5.2.2 GitHub

As our tools for version-control, we used GitHub. GitHub is a version-control service where we can work on different version of the same project locally on our own computers. This is incredibly useful to us when we're not in physical meetings and when we're working on different GUI element. When we need to synchronize our local work with the work of others, we can push our code to GitHub, which then handles the merging of our code into a single unified source-code.

[5] The online whiteboard for easy collaboration

5.2.3 Scrum

After discussing agile and plan driven development models, both within the group and with our client, we concluded that agile development is the best fit for our project. We had expected that we might have new ideas or new skill sets through the project and might then want to implement those ideas, which is the opposite of what the waterfall, incremental model or any of the plan-driven development models are.

We chose scrum due to previous experience with it on other projects and thesis while attending University. Scrum is a fantastic framework for our project since it's a great fit for a small dev team, because it splits goals into time-based segments called sprints, as well as helping us organize our current and future planning through the backlogs and sprint-logs. The use of backlogs and sprint-logs worked perfectly with our project since we were always only working on one button at a time, and because we were working within a short timeframe. Scrum can be hard to work with if you're working on a large project due to time estimation and workload. We used a Gantt-schema to help us both visualise the timeframe of each sprint, and to more clearly see the milestones and deadlines we had previously set up. This schema became especially useful whenever we were behind or ahead of schedule.

5.2.4 Kanban

Kanban is great for visualising and managing our workflow. Having the Kanban board up during the sprint meetings helped us see what tasks are finished or are still ongoing and was a great help in minimizing the amount of WIP (work in progress) we had. Each task's duration is tracked on the board through either an "in progress" or "ready" state. This also helped us improve our time estimation for future tasks.

6 Design and implementation

In this chapter we will be presenting the ways we implemented certain functionality and go in-depth in both the code and structure. The reader should obtain a clear understanding of how all the parts of the system work together, as well as how the functions responsible for a functionality works to deliver the expected result.

6.1 Wireframing

As mentioned in the previous chapter, during the first week we made a wireframe of our app on the website Miro. Miro has several templates for designing various things, including the application framework. The application framework was the base of our prototype, but we did not use most of the ideas in the framework for various reasons, such as: looking too much like a website, Tkinter limitations, being too complex compared to our client's wishes, among other issues. The reader will find screenshots of the wireframe we based out application from, and below that we will continue to show the finished GUI.

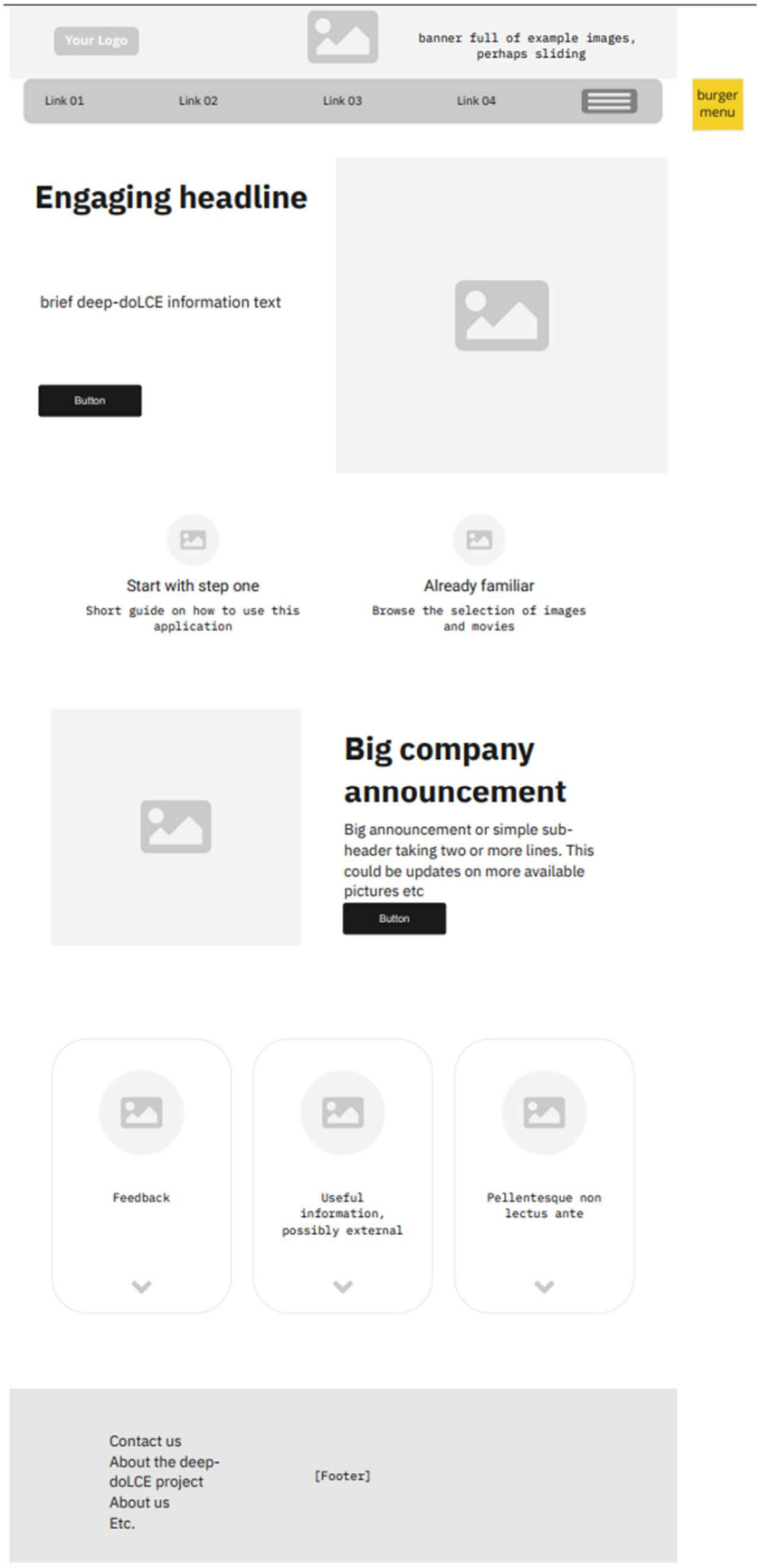


Figure 6-1 Wireframe front page

Figure 6-2 Wireframe categories page

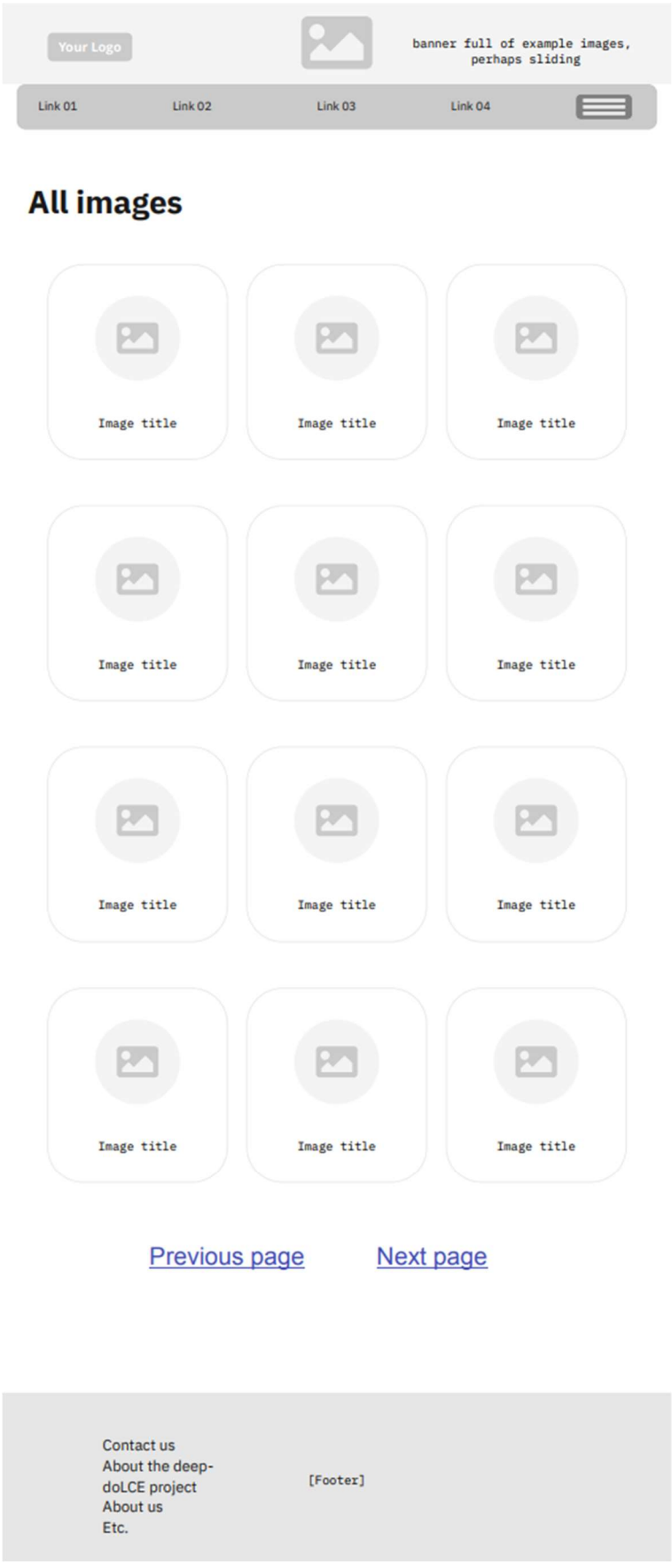


Figure 6-3 Wireframe image page

6.2 Programming the first prototype design

After choosing Tkinter as our GUI programming tool, we began to slowly program a design, we started with a random colour background, chose our logos, we did not like the button shapes that are available from Tkinter, because they looked dry and, so we found some images and edited them to program them into buttons, just so we could have an initial design we can work on upgrading with time.

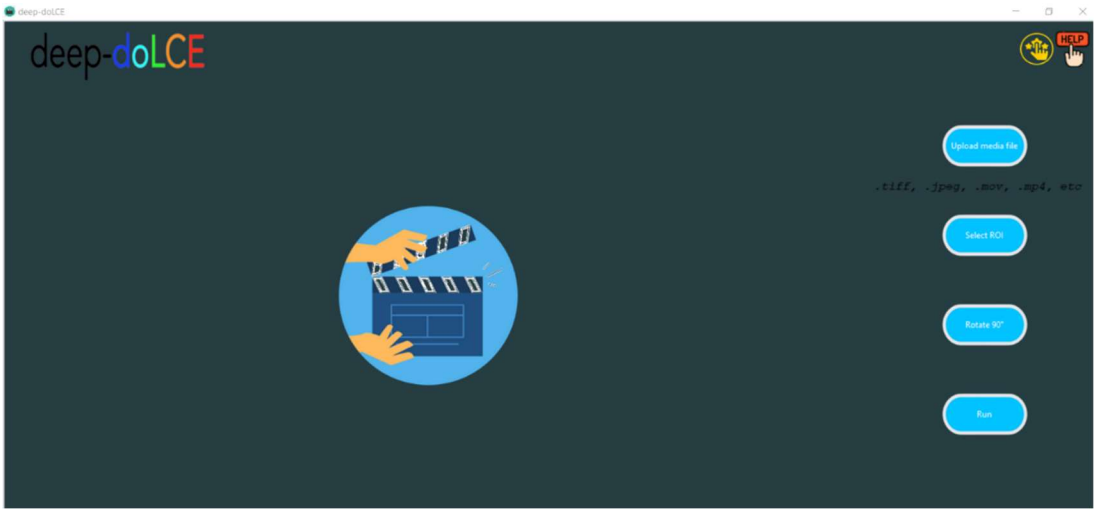


Figure 6-4 First programmed design

6.3 Choice of Colour

Choosing a colour was more difficult than we initially thought due to Tkinter limitations, which forced us to use a more minimalistic style. We found a colour online that we thought can fit into the designed theme and attract the user [*3*]. Picture underneath this text.

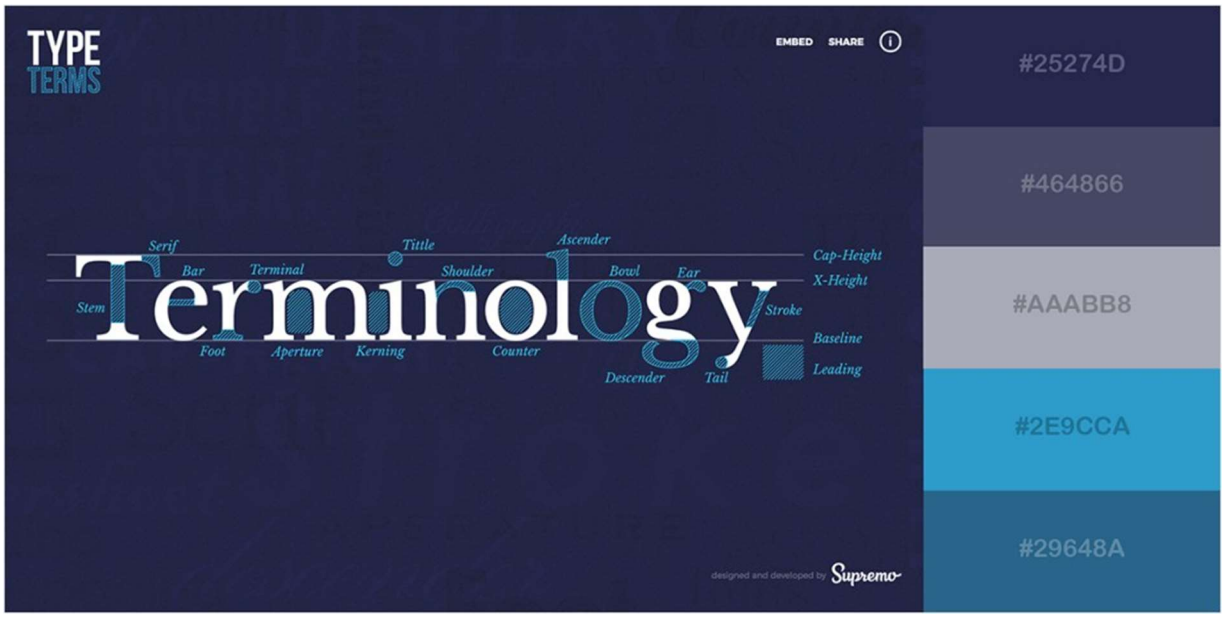


Figure 6-5 Theme colour inspiration[6]

[6] *50 Gorgeous Color Schemes From Award-Winning Websites*

6.4 Layout

Our final layout was a simple GUI with every button displayed on the side menu and header. The only change between buttons happens on the main content box, towards the left and centre of the screen.

6.4.1 Header

The header only contains the small logo of Deep-doLCE to the left, and a help button towards the right side. Pressing the help button brings up a brief explanation of what the application does, and the function of each button.

6.4.2 Side menu

The side menu contains the four buttons that are the main function of the application. These buttons are: The upload button, run, test, and select ROI. While most of these buttons are self-explanatory, we will still briefly explain their functions.

The upload media file button lets you upload the greyscale photo that is to be colourized. It possible to upload one image, a series of images or a video in any image or video formats.

The select ROI button allows you to select a region-of-interest to be colourized. You can select one area or multiple if you so choose.

The test button runs a quick test on the film to see that it gets the correct colours before you start the colouring process.

The run button confirms and starts the deep-doLCE code that colourizes the image(s).

6.4.3 Main content

The main content is the biggest section of the screen, taking up approximately 60% of the space. This section is where you will see the content pop up when you press a button. The press of a button removes a canvas and puts another one up instead with the desired action.

6.5 Pages

Below the reader will find screenshots of all the pages with a detailed explanation, ordered by the order of buttons in which they appear from top to bottom.

6.5.1 Main Page

The main page pops up when program is started and it includes the program logo on the top left, side menu buttons on the left that appear once needed, the middle area is an image of the program icon which changes depending on which buttons in the program are pressed. Three extra buttons appear when pressing the test button and disappears after testing is finished.

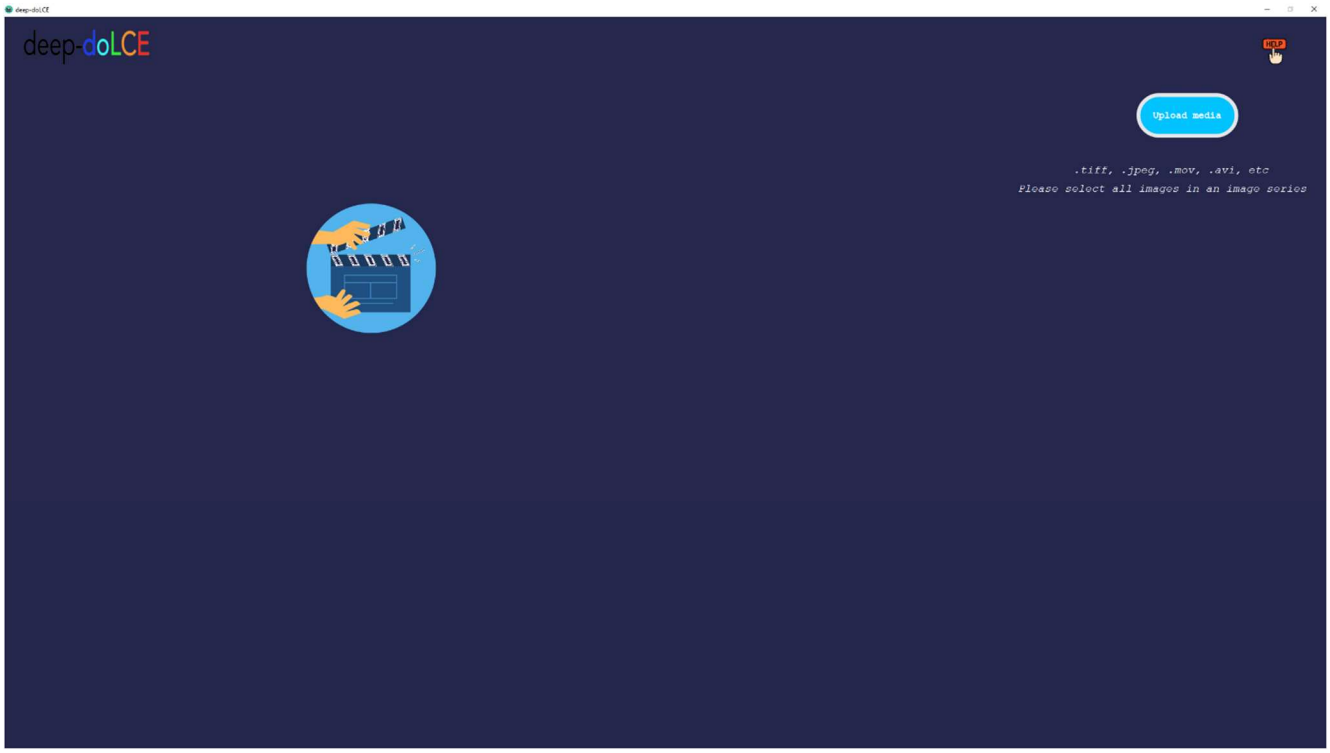


Figure 6-6 Main page

6.5.2 Help Page

The help page pops up when program is started and it includes the program logo on the top left, information about the program and instructions in the body and a “Back to main page” button.

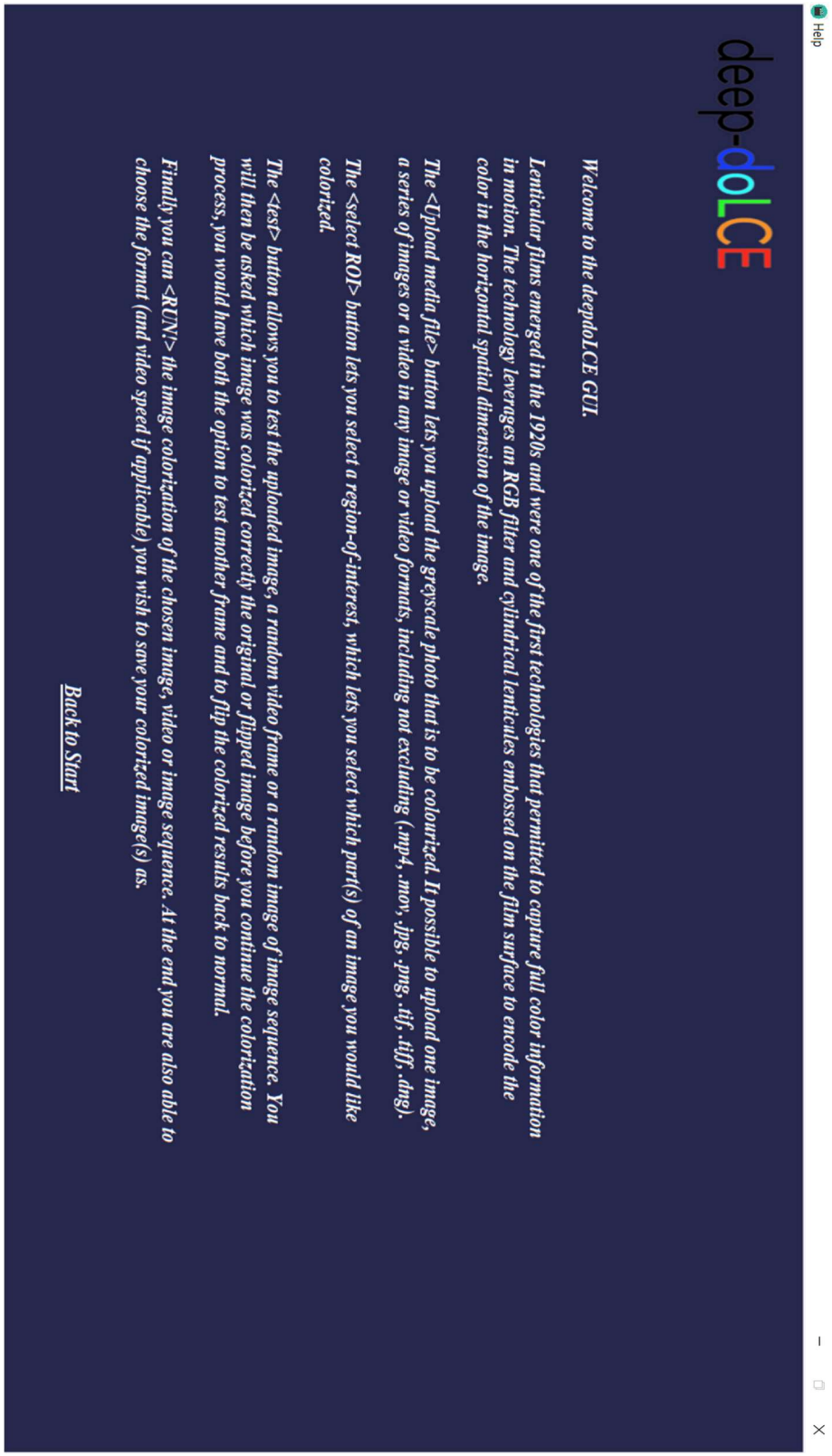


Figure 6-7 Help page

6.5.3 Upload Media page

The first frame of the uploaded files would be shown after uploading the image, as shown in the figure below.

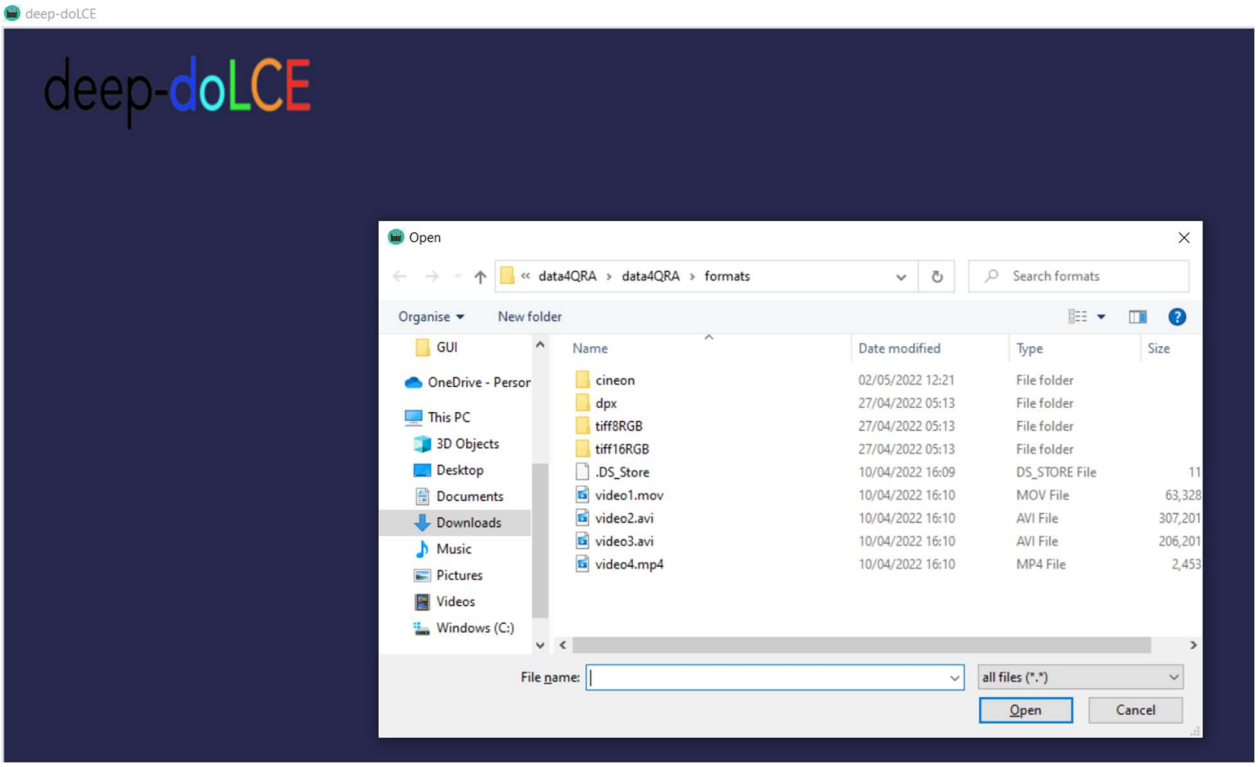


Figure 6-8 Upload media page

Figure 6-9 Upload media page

6.5.4 Crop page

The crop page pops up after pressing the “select ROI” button, it has a helpful tool bar in the top an image or a frame to select the region of interest from in the body.

Figure 6-10 Crop page

6.5.5 Test page

Shows a random colorized frame both original and another flipped before colorization, and it allows the user to pick which picture looks better to colorize all frames the same way, also allows the opportunity to test another frame.

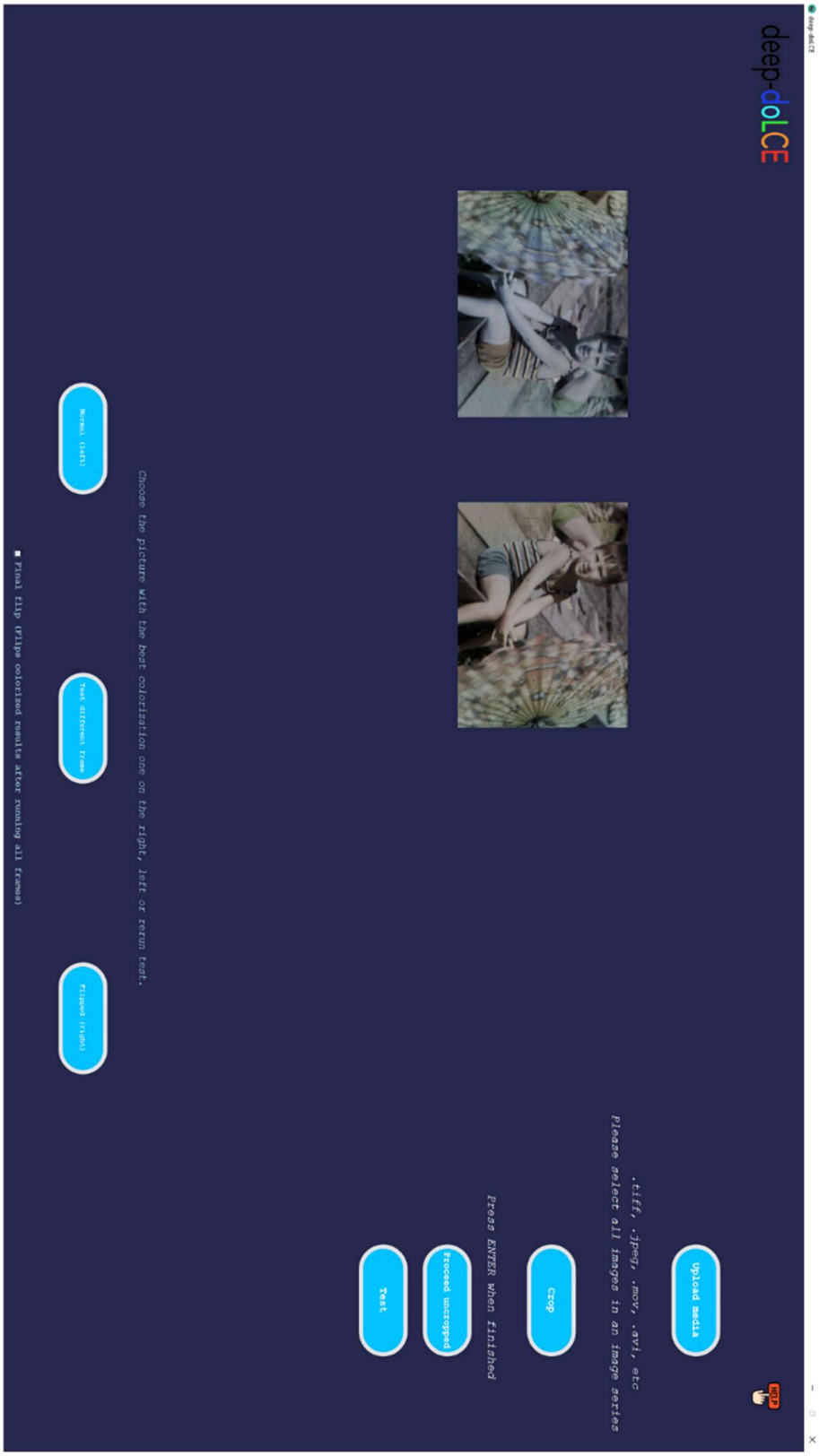


Figure 6-11 Main page after pressing test button

6.5.6 Run page

It allows the user to select master image series format and proxy video format, in addition to frame per second option for the video output.

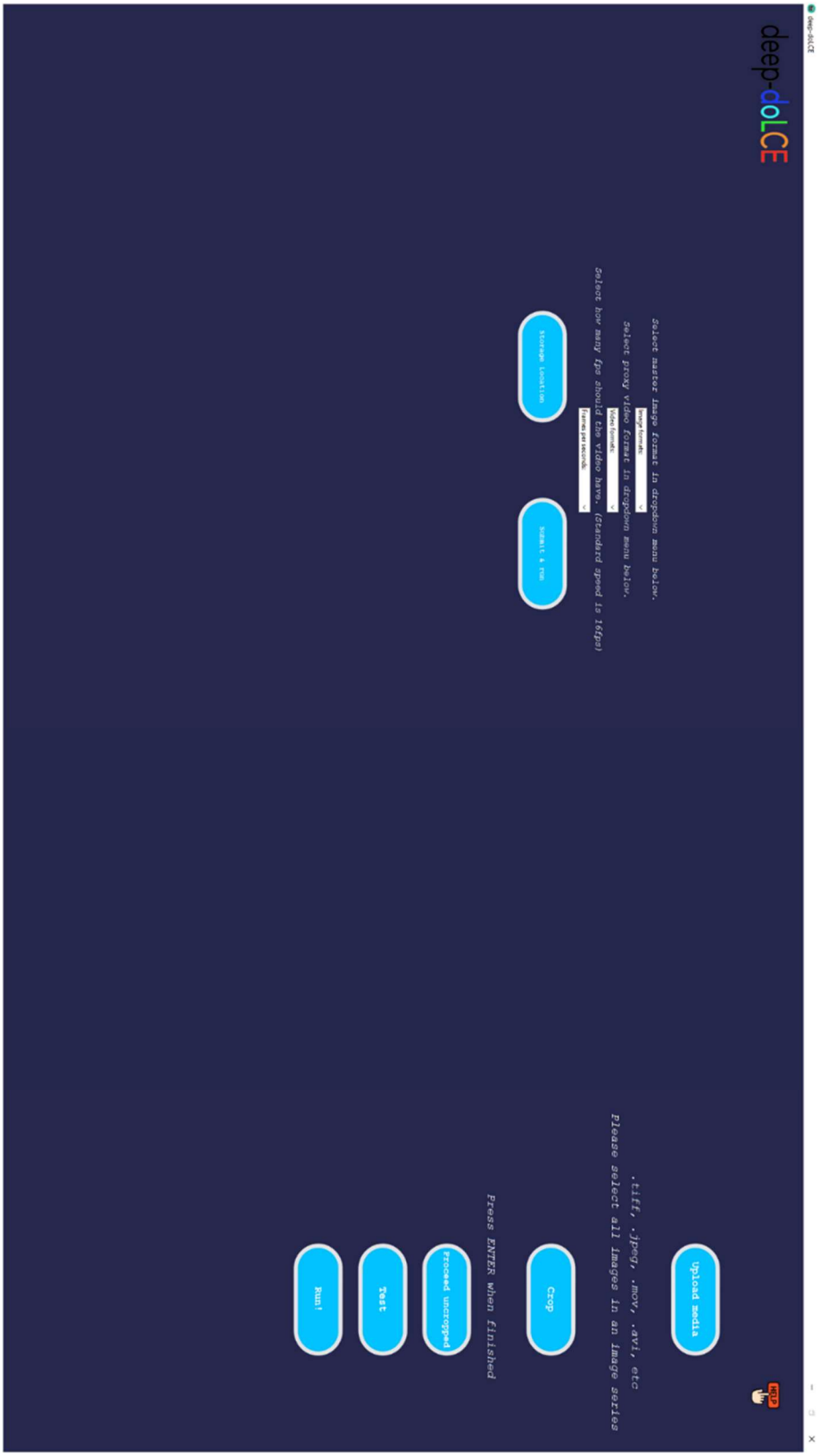


Figure 6-12 Run page

6.6 Programming the functionalities and backend of buttons

6.6.1 Help Button

The functionality behind this button is simple, we had to program a new window to pop up when pressed with the same colour and logo location as the main page with the help text in the body read from a text file and a back to main page button

6.6.2 Upload Media File

The functionalities we had to program behind the “Upload Media File” are three:

- Uploading multiple files:

We had to program the upload of multiple files to allow the user the ability to upload a series of images at once, using Tkinter’s * filedialog.askopenfilenames * function.

- Turning videos into image frames

Since the deep-DOLCE backend that was provided to us only accepts images not videos, each uploaded video must be turned into image frames, we did that through OpenCV.

- Turning images into 8-bit grayscale PNG

The deep-DOLCE backend that was provided to us only accepts 8-bit grayscale PNG so all video frames and images must be converted to that format. At first, we programmed that function by changing the 16-bit image, bit by bit but it took a lot of time to run, so we found NumPy code on stack overflow to turn 8-bit images to 16-bit images, so we reverse engineered it and it worked faster.

6.6.3 Crop

We programmed a select region of interest function using OpenCV, it takes the region of interest from one frame and crops all frames accordingly.

6.6.4 Test

The way test button functions is as follows:

- Chooses random frame from the folder

Given that lenticular films tend to have damaged or irrelevant frames at the beginning and end of the movie. Program chooses random frame in the middle of the movie by disregarding the first and the last 25% of the frames in the movie.

- Colorizes the frame and displays it

Lenticular films can format colour in RGB or BGR depending on which way you view the frame. Therefore, the test frame must be colorized and displayed in original and inverted form.

- Gives user ability to decide how to colorize the rest of the frames.

Some frames can lead to uncertainties as to which colour format is correct and if that's the case user can choose to run a test on different frame. If that's not the case user can choose which colour format is correct and whether it should be saved in original orientation or inverted.

6.6.5 Run!

The run button does the following:

- Gives user ability to choose output formats and where to save them

User can decide what image and video format to save colorized files as, as well as decide where to save them. After submitting the values, it runs through all the frames in the folder, colorizing as well as displaying them on the screen. Bar underneath the picture shows how much progress has been made with colorizing the entire folder.

6.7 Scrapped/Dropped ideas

This sub-chapter will mention the ideas we had from the beginning and the reason why we ended up not going with them. Some of these ideas never made it past the drawing board while others were worked on for several hours in coding and/or research.

6.7.1 Banner

One of the ideas we had while making the wireframe was to have a quick preview of the before and after of two images in the banner. Upon diving into Tkinter we found out that having an invisible background for images and canvases was not possible for both Windows and macOS[7], and so we would have to have separate versions for each, which we ultimately deemed unnecessary since: the client did not ask for it, the banner does not actually serve a purpose, and most importantly it might go against the main function of the app as a machine learning colour reconstruction tool. This would have been possible though through buttons with different backgrounds that end up displaying what looks like a one single banner. This would have looked like a normal banner but would cause bloat in the code and needlessly increased the folder size.

6.7.2 Day & night theme and increased font size

The issue here was that every code we found for real time updating of background colour and font size & colour did not work. We could instead have treated the GUI like two different pages and had the click of the button close the default GUI page and loaded in the day theme version, but this would cause problems if someone would change the theme while the colour reconstruction was underway. Another issue was that the user could only change the theme three times before the app closed itself. Ultimately, again, this was not something the client requested and could cause issues down the road if someone were to continue to work on the GUI in the future.

6.7.3 Rate Button

Lastly was the rate button. The client had (as far as we could figure) talked about having a rating or evaluation by his peers and the team mistook that as a button to design. This was a simple misunderstanding between the team and the client which led to the scrapping of the button as soon as we brought it up again with the client. Below is a screenshot of the small progress we had before scrapping the button.

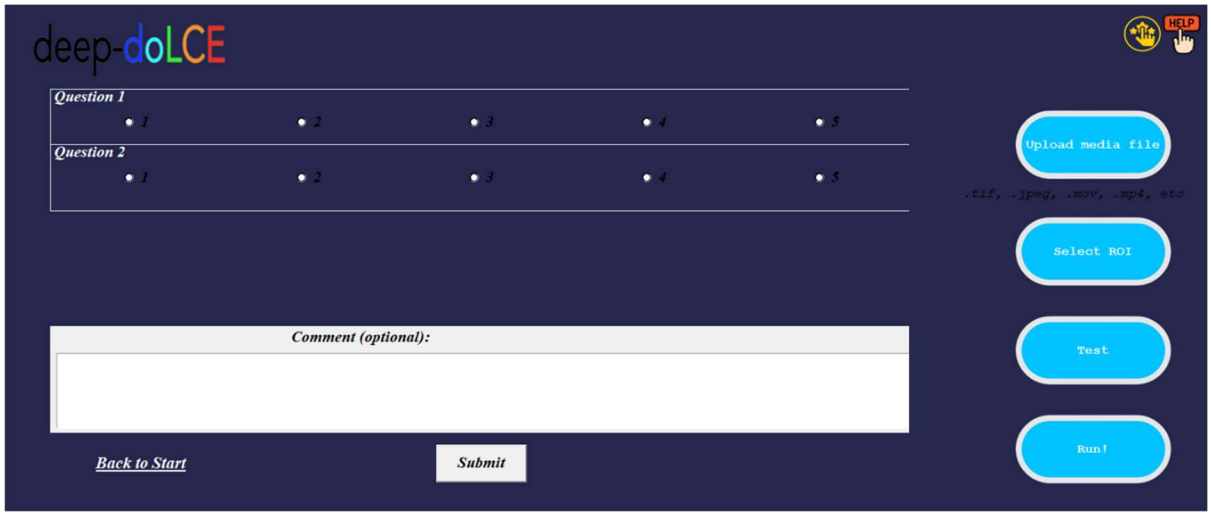


Figure 6-13 Rate button page

[7] Transparent background in a Tkinter window

7 Choice of the development environments

In this chapter we will go over the environments and tools we used during the development of our application. The chapter is short and split into two since we were only developing a front end, the IDEs' and repository.

7.1 IDE

Our choice of IDE came down to each individual team member since we would mostly be coding in separate locations. Everyone used the IDE they were most experienced and comfortable with. The IDEs' chosen were Visual Studio Code and PyCharm.

Visual Studio Code, or VS Code, is a source-code editor developed by Microsoft for Windows, Linux and macOS. It lets you easily install multitude of extensions/plugins you want or need to speed up the development process.

PyCharm is an easy-to-use python specialized IDE with many python tools and packages that are easy to use and the option to choose python interpreter that would come in handy with making the program work on GPU instead of CPU.

7.2 Git

In order to share the code, we chose Git to both easily pull and push our code to our central repository on the cloud service GitLab. This helps keep our code consistent and prevents any errors due to changes made before merging the code. This is accomplished by first pulling the source code on GitLab and merging it with the code you have written and are attempting to push/upload. You then resolve any merging conflicts and then Git lets you, most likely, successfully push/upload the code into the chosen branch, where it becomes the new source code. GitLab also allows you to roll back any changes if you so choose, through its archived history system.

8 Code Quality

Code quality is not only extremely important to us as developers but also for any future development process of the application. This means that our code had to be readable and understandable by anyone with even very little coding experience and we set out to do just that.

8.1 Code review

Since most of the time spent developing was through separate, individual work, it was important that we could all understand each other's code.

8.2 Code comments

Comments are crucial so other developers can continue working on advancing this project both frontend and backend, so we made sure that comments were added and that our code is easy to read.

9 Testing

In this chapter we will talk about the testing we did and the improvements it led to.

9.1 OS compatibility

As mentioned in the previous chapter, it was extremely important that the application would run on both Windows and MacOS. Since we developed the GUI with that in mind the entire time, when it came down to testing it on both OS, we found that it worked perfectly on both. However, we did not test the application on mobile devices since the client did not ask for it, and because of the computational requirements being presumably too high, which could damage the device.

9.2 Acceptance tests

One of the most important and last tests for applications is the acceptance tests. Our application runs almost exclusively using buttons, with the only exception being the Upload Media button. This means that when the testing week came, testing the whole application was quick.

9.3 Usability Tests

Usability testing is the practice of testing how easy a design is to use with a group of representative users. This type of testing is one of the most important parts of the project for us because it allows us to see how the average future application user interacts with the software and what they think should work differently. These tests consist of different use-case scenarios that the users will complete while the observers observe and take notes. It can be important that the user-group has a wide range of knowledge and experience with similar systems.

There are many different types of usability tests, and we chose remote unmoderated assessment testing through Zoom & Team Viewer, where the users would remotely access a computer at NTNU that had a good GPU. Users would then commence testing of colour reconstruction with minimal input from the observers, but the observers were always there for any questions and/or explanations.

While the software is designed for everyone, both those experienced with lenticular films and not, the user-group we had chosen with the client from the beginning was a group of highly experienced film archivists. However, they would also try as best they could to act like they knew nothing about lenticular films or images and films in general.

10 Discussion

This chapter will talk about the end results of our software and alternative solutions we had after talking to experienced film archivists during the usability testing.

10.1 Results

10.1.1 Speed of colour reconstruction

We have managed to make the backend provided to us work faster, by making it run on GPU instead of CPU, by installing anaconda and using Conda Python environment, we faced many problems doing so, like problems with attempting to install some Conda packages that were not compatible with the python version used, as stated in their developer logs. Another problem we faced is that we needed a stronger GPU than the ones we had on our laptops, so we had to find a workstation with a strong GPU with an 8GB memory to conduct our usability tests. Other specs of the pc were not given to us as we had also neglected to ask. Normally while running the program on CPU, it takes a minute to colorize a frame, on the other hand using GPU it takes around 12 seconds to colorize a frame.

10.1.2 Converting an image from 16-bits into 8-bits

One of challenges we faced was to turn a 16-bit image into an 8-bit image. The reason why we had to convert the images was so that we could process the images in Giorgio's code, as it only accepts 8-bit images. All the code we found online that did the conversion would do it by going over small sections at a time and individually turning them into 8-bits, resulting in slow conversion speeds. Giorgio got us in contact with someone that knew the ins and outs of NumPy that helped guide us to a code that efficiently turned 8-bit images into 16-bit. We reverse engineered that code to do the opposite, resulting in much faster conversion time.

10.1.3 Flipping the image and choosing another frame

The test page has three important buttons after the user has colourized the random frame as previously mentioned in sub chapter 5.6.4. The client and the team had a discussion about automatically saving the final image or image sequence in its original non-flipped rotation. However, the client brought up the point of inverted text, where the user would have to then use another application to flip their new film so the text can be in the right orientation. Instead, we let the user decide for themselves. An example of this can be seen in image 5-11.

10.1.4 From Select ROI to Crop

One of the task requirements was to create a button that selects the region-of-interest or ROI for short. While we, the team members and client, are all familiar with the term it's likely that others are not, especially if they're not fluent English speakers. We decided to therefore change the name of the button to Crop as we all agreed that it was a more common term.

10.1.5 Failed colour reconstruction

Since image reconstruction of a lenticular film is a precise process there are a lot of factors that can result in a failed reconstruction. An image can fail to be colour reconstructed for multiple reasons that are given below.

Image is misaligned. Deep-doLCE has observed a maximum of rotation of about 1° [8], meaning that the scanned image needs to be almost perfectly aligned because of the way the RGB filters are overlayed on the image.

The image is not a lenticular film. Our application is designed for everyone to use regardless of film archiving experience. Lenticular films are rare and requires an expert eye to identify whether the film is a lenticular film or not. A potential user might have found an old film and wonders if this might be a lenticular film and attempts to use our software since they're unable to see the difference between lenticular films and other greyscale films. The user can then attempt the colourization on our software.

Smeared or damaged parts on the image. A smeared or image will lead to issues with the visibility of the diagonal edges of the film. If this is an issue with only a handful of the films frames then our software will likely circumvent the issue by, as mentioned in testing chapter 5.6.4, testing random frames between the first 25% and the last 25% of frames.

Low resolution. A scanned image needs to be scanned in a high resolution (at least 4k) for the diagonal edges of the film to be clear enough for colourization.

Most of the above issues stem from the physical frames being damaged or issues with scanning, which are out of our control. It is therefore important for us to convey this information to the user without being condescending and putting the blame entirely on the user because there are of course issues that are on our end. This should be a pop-up box that appears after around half a dozen "test another frame" button clicks, preventing the user from wasting their time on attempting to colourize something that is impossible. Further testing of the product should help us iron out the issues caused by our software as they appear.

10.1.6 Failed video conversion

Library used for video conversion is a Python FFMPEG wrapper in OpenCV. It takes the frames and loads them into the video files one by one. During the conversion process some frames may be skipped for variety of reasons, for example:

- Complex codecs with not enough computing power can lead to PC not having enough time to write images into the video.
- Rounding errors when choosing FPS.

Choosing the output format can be hard if required codec is not available, especially when documentation for codecs is lacking. For instance, in FFPMEG following codecs are not available; H.264, H.265 or H.266. Yet without going deep into comments you would think H.264 is possible.

Lastly there are times when FFPMEG will throw errors on certain codecs even though it run just fine before.

[8] A Deep Learning Approach for Digital Color Reconstruction of Lenticular Films

10.2 Alternative solutions from usability testing

Below the reader will find the major bullet points we got from the two usability tests we did, which also mention which ones are already fixed or not. We will also discuss some of the bullet points in the following sub chapter.

10.2.1 Bullet points from usability testing

- Change location of buttons from left to right and put them as close to under the image as possible instead of far away. (User preference)
- Test another image button should be further away. Perhaps make the images themselves a button that you click instead of a button.
- Gray out or disable buttons if uploaded aa file was not uploaded already. (fixed)
- Buttons while testing should be disabled during the colour reconstruction. Selecting the ROI while image processing was ongoing crashed the program. (fixed)
- Limit selection of frames per second options to 24, 16, or 1 and nothing more. Mention in the application that standard speed for films at the time was 16 fps. (fixed)
- Area of image can be bigger upon uploading.
- While uploading files, if you do not click an image inside a folder then all images inside that folder get selected for image sequencing.
- Should we restrict the file formats further, MP4 and similar formats might have heavily compressed files that might not get processed. (fixed)
- General input: straightforward. For professional use, wouldn't bother with video outputs and would prefer proper DPX format so he can do whatever he wants with it. Would personally just use .tif, and H.264 .mp4. Archives would not want .jpg, .mov, or .avi.
- If the user picks a different frame during the "test" process of an image sequence several times (4-5 times) a window should pop up asking "are you sure the requirement of the scan is met?"
- Upload media button size for images is small due to being a dynamic sized window. (fixed)
- Output files should have the same names as the original but with underscore and coloured or some other variant (colour_img001.tif etc.)
- Instead of cropping the image, keep it as a region of interest and run the colour reconstruction process only inside that region, while leaving the rest of the image untouched but keeping that as part of the image (don't throw away the black & white edges/frame).
- The video file is what's called proxy. Proxy means a low weight file for visualization purposes only and the image sequence will be called master image sequence. Master means good for further processing and to have the best quality necessary. This will have to be somehow clear in the run page

10.2.2 Feedback discussion

The usability testing taught us a lot about film archiving and showed us how important it is to get feedback from professionals. Below are some of the noteworthy issues and discussion points that came from the feedback.

Our application before the feedback had the image sequencing frame rate options of 1, 24, 30 and 60. It was brought to our attention that the frame rate at the time was most commonly 16 and film archivists will most likely only ever want to pick 16 fps, or the standard 24 fps used in film making. We also chose to include 1 fps in case of a smaller image sequence or if the user simply wants a

lower frame rate for their file. Without the feedback we would never have known about the 16 fps, and we would have included frame rates that no one would want to use because it would result in unnatural movement.

One of the general potential issues that were discussed was our choice of allowing the user to save their colourized image in any of the common formats currently possible. This issue is multi-layered with clear pros and cons. The film archivists argued that .jpg, .mov, and .avi will never be used by film archivists and could even lead to other archivists scoffing at the software and not using it because including those formats is unprofessional in the world of archivists. On the other hand, the average person might recognise .jpg and .mov the most and might get confused by only the choice of .tif and .mp4 files. The question then was yet again, who exactly will be expected to use this application the most? We currently settled for including the option of the "unprofessional" file formats since it might be better to include options rather than remove them. This could be changed in the future, especially with some user data.

Sometimes both the flipped and pre-flipped images will turn out bad or uncoloured. When this happens during an image sequence the user can choose to test another frame of the sequence. The suggestion given by Giorgio was to have a window pop up that asks the user: "are you sure the requirement of the scan is met?", in order to make sure the user does not waste their time on something that will simply not work. A failed colour reconstruction can happen for multiple reasons, for example because; image not aligned correctly, the image is not a lenticular film, the film has some damage, or because of a low resolution.

One of the film archivists suggested an improvement for the final result image that could be highly desirable for archivists. Archivers would prefer to keep all image information since sometimes there's interesting information on the edges that helps film archivists. This suggestion was then to run the colour reconstruction on only the cropped part of the image, yet keeping the whole image intact, resulting in a coloured image in the chosen region-of-interest with black & white edges/frames. Archivists might be very interested in this function because it might help them, while the average user might be interested in this function for its artistic aspect.

11 Conclusion

This chapter discusses the work we did, feedback from the client, the future of the project and what the project taught us.

11.1 Summary of the work done

The goal of this project was to create a GUI for deep-doLCE that is usable by not only people with experience in film archiving, but by anyone who might have stumbled upon a black & white image or film and wants to check if this is perhaps a lenticular film. In order to meet these criteria, it was important to have a simple and intuitive GUI that anyone, regardless of age or experience, could use.

This led us to design a system that would only display two buttons upon starting the software, the "Help" and "Upload Media File" buttons. After pressing the "Upload Media File" button two other buttons, Crop and Test, will show up and hopefully it is clear to anyone what those buttons do. After testing the image or image sequence and confirming which image got colourized correctly, the run button appears, which lets you choose the format you would like your new colour reconstructed image or image sequence to be in.

When it came to the colour scheme of the application, we chose a clean design based on the colour theory. In the beginning of the project, we had a grand idea of a banner displaying some previews of colour reconstructed images and a more colourful background, but due to Tkinter limitations this idea was not realized. Yet we were happy with the results and even praised during the usability tests for a clean and simple GUI.

The GUI had to connect to the backend given to us by the client, Giorgio. Our group had to use a python GUI framework to develop the GUI since that was the language the deep-doLCE project was developed in. The framework we chose was Tkinter since it was it is one of the most popular GUI frameworks and was easier to install than our other top option, Kivy. With Giorgio's help were we then able to find which functions inside the backend we would be using.

Deep-doLCE currently only accepts 8-bit images, which meant that we had to turn 16-bit images into 8-bit images. We found NumPy code on stack overflow that turns 8-bit images to 16-bit images, so we reverse engineered it to turn 16-bit images into 8-bit images which worked perfectly on all the images provided by Giorgio but also random images we found online. This code will be obsolete in the near future when deep-doLCE will be changed to accept 16-bit images by default which will improve the image quality.

One of the biggest contributions we made was by making the backend run on the GPU rather than the CPU, which sped up the colourization process immensely. This was one of the most time-consuming parts of the project and posed great challenges and required a lot better GPU than our laptops have.

Essentially what we have done is to create five main buttons, 'Upload Media File', 'Crop', 'Test', 'Run!', and 'Help'. Behind the scenes we have also turned any 16-bit image into an 8-bit image so the deep-doLCE backend code can colourize it, and then ran that backend code on a computer's GPU rather than CPU.

11.2 Future Work

While we did everything the client requested and more, this software will see a lot of improvements in the future by the client and his team. Some of these will be made because of improvements in the backend, while some are from ideas that were had later into the project that the team did not have time to implement, and some ideas that were not had in any of our meetings. The client mentioned

plans to have an intern change the backend to process images in 16-bits, which will mean that our code to change an image to 8-bits before colourization will need to be changed.

As mentioned in the previous chapter there are already many ideas for improvements such as unique features that only run the colour reconstruction on a select part of the image but leaving the rest untouched, or QoL changes like the automatic file name, not selecting any image in a folder should select all images by default, and a feature that turns the resulting images during the test process into buttons.

It is worth noting that the time it takes to reconstruct the colouration of the films will continue to drop as technology improves. It might even be possible for phones in the future to run this software, but changes will have to be made to the code so that Android or iOS can run it.

11.3 Learning Outcomes and Concluding Remarks

Prior to this project only one person had previous experience with making GUIs with python, but we can confidently say that everyone learned a lot about it during the past four months of work. We learned relatively simple things such as creating texts and buttons and attaching functions to them. We learned how to connect the buttons to the backend. We learned how to efficiently turn a 16-bit image into an 8-bit image as well as turning a video into image frames. We learned how to make a software run on a GPU instead of the CPU.

The project not only taught us a lot about designing a GUI through a python framework, but also about the existence of lenticular films and how they work, as well as image and film archiving in general. Overall, we come out of this project with a lot of useful experience for our future careers as well as just interesting new knowledge about the world of film archiving.

Bibliography

*[1] Dr. Giorgio Trumpy, and Dr. David Pfluger, "*deep-doLCE: A New Machine Learning Approach for the Color Reconstruction of Digitized Lenticular Film" University of Zurich, 27 May 2021, https://www.film.uzh.ch/en/research/projects/verbund/deep-doLCE.html *Accessed February 1, 2022.*

[2] Cambridge in Colour, "UNDERSTANDING CAMERA LENSES" Cambridge in Colour [website], [no date]. https://www.cambridgeincolour.com/tutorials/camera-lenses.htm Accessed May 19, 2022.

*[3] Stefano D'Aronco, Giorgio Trumpy, David Pluger, and Jan Dirk Wegner, "*A Deep Learning Approach for Digital Color Reconstruction of Lenticular Films" Researchgate, 10 February 2022, *https://www.researchgate.net/publication/358579081_A_Deep_Learning_Approach_for_Digital_ColorReconstruction_of_Lenticular_Films Accessed May 16, 2022.*

[4] Miro, "The online whiteboard for easy collaboration" Miro [website], [no date]. https://miro.com/online-whiteboard/. *Accessed February 3, 2022.*

[5] N. Chibana, "50 Gorgeous Color Schemes From Award-Winning Websites" Visual Learning Center [website], September 4, 2016. https://visme.co/blog/website-color-schemes/. *Accessed February 8, 2022.*

[6] forumfresser, "Transparent background in a Tkinter window" Stackoverflow [website], September 29, 2013. https://stackoverflow.com/questions/19080499/transparent-background-in-a-tkinter-window *Accessed March 28, 2022.*

Appendix A. Project Plan

NTNU

Bachelor Thesis
GUI 4 deep-doLCE

Project plan

Runar Astvaldur Hedin, Aleksander Kokowski, Qeis Khaled Qeis

January 2022

NTNU

Norwegian University of Science and Technology

Contents

1 Goals and frames

1.1 Background

"Some of the first home movies in color were shot on 16 mm lenticular film during the 1920s to 1940s. This very special film is embossed with a vertical array of hundreds of tiny cylindrical lenses that allowed to record color scenes on a black&white silver emulsion. The most efficient approach to obtain digital color images from these historical motion pictures is to scan the silver emulsion in high-resolution and let a software extract the encoded color information. The localization of the lenticular screen is the first and most complicated step of this process. A 'classic' signal processing method proved to deliver successful results in some cases, but more often adverse factors—damaged or warped film, scanning problems—hinder the successful localization of the lenticular screen.

The deep-doLCE project explores a more advanced and robust method, using an already available big dataset of digitized lenticular films to train a new deep learning software. The aim is to create an easy-to-use software that revives awareness of the lenticular color processes thus making these precious historical color movies available again to a public and securing them for posterity." [1]

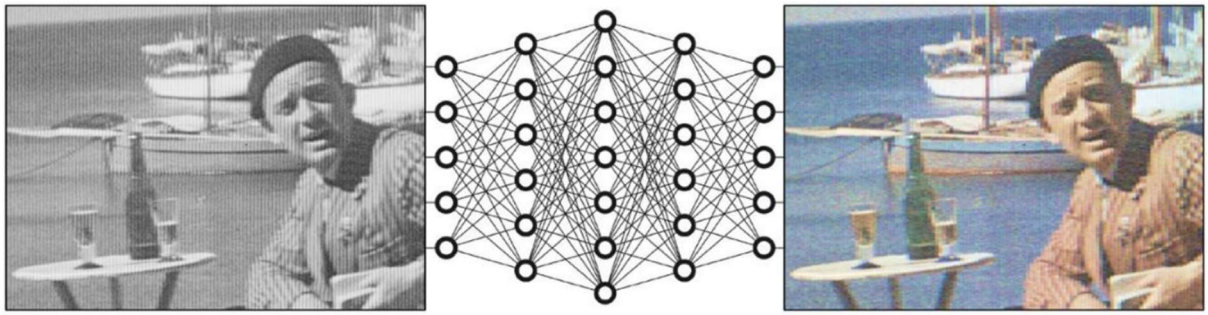


Image 1, Trumpy & Pfluger *[1]*

1.2 Project Goals

The goal of the thesis is to design and test a user-friendly Graphical User Interface for deep-doLCE.

1.2.1 Result goals:

Outcome goals are the goals describing the wanted result from the project

- Design a GUI for the deep-doLCE project.
- The application should allow you to select an image or movie from the database, that the deep-doLCE project code then processes and colorizes.
- The application should offer the user the possibility of selecting a Region-Of-Interest (ROI).
- Should be able to launch the color reconstruction process and monitor its progress.
- The application should be easy to use, with a clean and simple design.

1.2.2 Effect goals:

- Easier to select images and movies to colorize.
- Improve compatibility of deep-doLCE with the most common file formats used by film scanners.

1.2.3 Learning objectives

Learning objectives is the knowledge and experience we hope to gain during the project.

- Gain experience on frontend development.
- Gain experience on backend development.
- Gain knowledge and experience with various file formats.
- Gain more experience with database/s.

1.3 Project frames

The bachelor thesis has a specific timeframe. The project starts 11. January 2022, with a due date of 20. May 2022. The software should preferably be finished by 20. April and will be handed in, along with the thesis, by 20. May. Deep-doLCE is written in Python, so the GUI must be written in PyQt or other available tools for Python, such as Tkinter, wxPython or JPython.

2 Scope

2.1 Problem area

As Deep-doLCE application code exists, the application lacks a graphical user interface (GUI) that simplifies and makes using the application simpler. The GUI should offer the user the possibility to select a Region-Of-Interest (ROI). The existent Deep-doLCE application code should be improved to support more input and output format options.

2.2 Task description

In order to add a functional GUI, that offers the possibility to select a (ROI) to an improved Deep-doLCE application code that supports more input and output format options, the following steps should be achieved:

1. *Identify the most suitable tool and start designing the interface*
1. *Include the buttons that let navigate the file system and select input files and destination folder*
1. *Improve compatibility of deep-doLCE with the most common file formats used by film scanners*
1. *Offer to the user the possibility to select a Regio-Of-Interest (ROI)*
1. *Let the user choose the file format for the output, including a down sample option.*
1. *Run a test reconstruction on a single frame (a pre-flip left-right could be necessary)*
1. *Create the RUN button that start the reconstruction of the whole movie*
1. *Give the user feedback on the status of the process (i.e., output image, ETC, etc.)*

Additional tasks:

- *Expand the data augmentation with more transformations (e.g., blur, barrel/pincushion distortions)*
- *Allow for small curvature in the vectorization of the network's output*
- *Create a user guide to the application*
- *Add an option to review the software and give feedback.*

3 Project organization

3.1 Responsibilities and roles

The team chose Qeis Khaled Qeis as the project leader. The project leader is responsible for organizing meetings, contacting the client and the university staff, and monitoring the group progress so the schedule is followed.

The team also chose Runar Astvaldur Hedin as the project recordkeeper, which has the responsibility of taking backups of documents, writing minutes of meetings and document important decisions.

Proofreads and submits each document/thesis (quality assurance for the code will be a shared responsibility for the entire group). Aleksander Kokowski will be responsible for that role.

3.2 Routines

Every group member is expected to work as hard as possible to stay a week ahead the project plan (around 30 hours weekly). The group has scheduled weekly meetings with both the university supervisor staff every Monday 10:00-11:00 (extra meetings can be arranged when needed), meetings between members and the client can be arranged when there is need for them or to show

progress (at least once every two weeks). Meetings should be scheduled in advance at a time that everyone agrees on and fits everyone's schedule. Due to the COVID19 pandemic and work schedules, these meetings will be held online as much as possible.

3.3 group rules

To ensure the groups efficiency and effectiveness is kept, we decided to implement group rules applying to every member.

Absence

Every member should and is expected to attend all meetings. If this is not possible the group member should let the other group members know a minimum of 2 hours before the meeting, with an exception for emergencies.

Effort

Every group member is expected to work as hard as possible to stay a week ahead the project plan (around 30 hours weekly including meetings and collective work). Work hours will be documented using Toggl.

Conflicts

In the case of an internal conflict where two or more group members disagree on an issue, the whole group should discuss this and come to a solution by discussion. A vote should take place if no mutual solution can be found.

Consequences of breaking the rules

In the case of a member does not show up regularly in meetings, breaks the team rules, or fails to do their tasks, then that member will be given a written warning and the group will collectively try to solve the issue. If the problem persists then the university project supervisors might be contacted.

4 Planning, follow up and reporting

4.1 Choice of system development models

After discussing agile and plan driven development models, both within the group and with our client, we have concluded that agile development is the best fit for our project. This is due to the project not only including the main goal, but also extra tasks if we have the time and/or picked up the skill set required to implement the clients extra task wishes. The finished product will likely be far more than creating a GUI. This is the opposite of what the waterfall, incremental model or any of the plan-driven development models are.

Out of all the agile development models we have chosen to go with a combination of SCRUM and Kanban. SCRUM is a fantastic framework for our project since it's both a great fit for a small dev team, and because it splits goals into time-based segments called sprints. We plan to have several weeklong sprints and 2 two weeklong sprints, with SCRUM meetings at least three days a week. One of these days will usually be used for the sprint planning, a short meeting at the start of each sprint, detailing what needs to be done before that sprint is finished. We will also have another sprint meeting at the end of each sprint for a sprint review and retrospective.

Kanban will be great for visualising and managing our workflow. Having the Kanban board up during the sprint meetings will help us see what tasks are finished or are still ongoing and is a great help in minimizing the amount of WIP (work in progress) we have. Each task's duration is tracked on the board through either an "in progress" or "ready" state. This could also help us improve our time estimation for future tasks.

4.2 Main division of the project

The project will begin with us getting familiar with the deep-doLCE code. We will need to get an idea of how the program works and which parts of the code we will need to run/interact with.

Our project will begin with developing a use case diagram, wireframes, and a high-fidelity prototype for the front end of our application. We will have to develop a database, but at this time we are unsure of how much this will end up being since we are working with a previously developed project. Lastly, we shall write the report, which will be based on everything we have worked on for the project itself and will contain our documentation of everything we did and learned during the project.

4.3 Plan for status meetings and decisions in the period

We will have a weekly meeting with our supervisors to discuss our progress, our problems, feedback, and to help us in case we need it and if possible. We always plan to finish each weeklong sprint before the meeting, even though each sprint should be finished the day after. The two weeklong sprints will have the option of getting feedback while the sprint is ongoing. We will also have a bi-weekly meeting with our client to discuss our progress.

5 Quality assurance

5.1 Standard tools

For the project, we have a large toolbox that we will be used to keep us as efficient and organized as much as possible. Some of the tools that we will be using are:

- Overleaf: is used for drafting the report and project plan with precise structure and layout.
- Teams: is used for meetings and a place for sharing documents.
- Mira: is used for designing wireframes.
- GitHub: is used to share the code between group members.
- Trello: is used to have an oversight over what tasks need to be done and their deadlines.
- Toggl: is used to selectively log time spent working on a single task.

5.2 Risk analysis

Below is a table showing the project risks and their likelihood and the possible consequences they have. The probability is divided into 4 cases: low, moderate, high, and very high. Consequences and risks are divided into insignificant, tolerable, serious, and catastrophic.

Number	Risk	Probability	Consequence
1	Project is not completed before deadline	Moderate	Catastrophic
2	One team member gets sick and is absent an extended amount of time.	Low	Tolerable
3	Loss of data, code, or report.	Low	Catastrophic
4	Supervisor is not available as needed.	Moderate	Serious
5	Client is not available as needed.	Moderate	Tolerable
6	Lack of documentation	High	Serious

5.3 Contingency measures

These are the contingency measures that are planned to decrease the probability of the risk happening, as well as lowering the aftermath of risks after they happen.

Number	contingency measures
1	Have a realistic project plan that is possible to follow and use tools like Trello and Toggl to follow up on time used on every task. Inform the product owner If a delay should happen and discuss if removing some planned functionality is an option.
2	Being careful as much as possible and avoid unnecessary physical interaction due to the covid-19 pandemic, reviewing each other's work to keep up to date and be able to take another member's tasks in case of illness.
3	The code will be in a Git repository and downloaded to keep backups. The report will be written on overleaf and saved online on teams, the project leader is responsible to download copies of the report for back up.
4	Weekly meetings with the university supervisors are planned. If the supervisors were unable to attend these meetings, contact them and ask for new meetings or have a meeting with the one available supervisor. If both supervisors do not attend any meetings, NTNU would be contacted.
5	A meeting with the client should be arranged once every two weeks, but if meetings cannot be scheduled with the client, then contact can be made temporarily by e-mail. If the client does not attend these meetings and are not able to reschedule nor able to answer our e-mails, the project members will make assumptions about the product and the client's preferences.

6	A reminder of documentation tasks will be created, as it is important to get proper documentation on the project.

6 Plan for execution

6.1 Activities

The Gantt schema is divided in to three main parts: Planning, Project Development and Thesis writing.

The first part, Planning, is dedicated to writing and signing the appropriate contracts, and creating the project plan.

The second part is divided into eight sprints of variable lengths estimated by the group, based on what Dr. Giorgio Trumpy had requested, and with a test phase of the application at the end. The list will be found at the bottom of this page.

The final part revolves around the actual Thesis and is split into 2 due dates. The thesis will be written slowly along with the project itself and we have been recommended to have the first draft ready a month before the due date i.e., 20. April, and be finished by 20. May. The second part of the thesis writing is preparation and performance of the presentation.

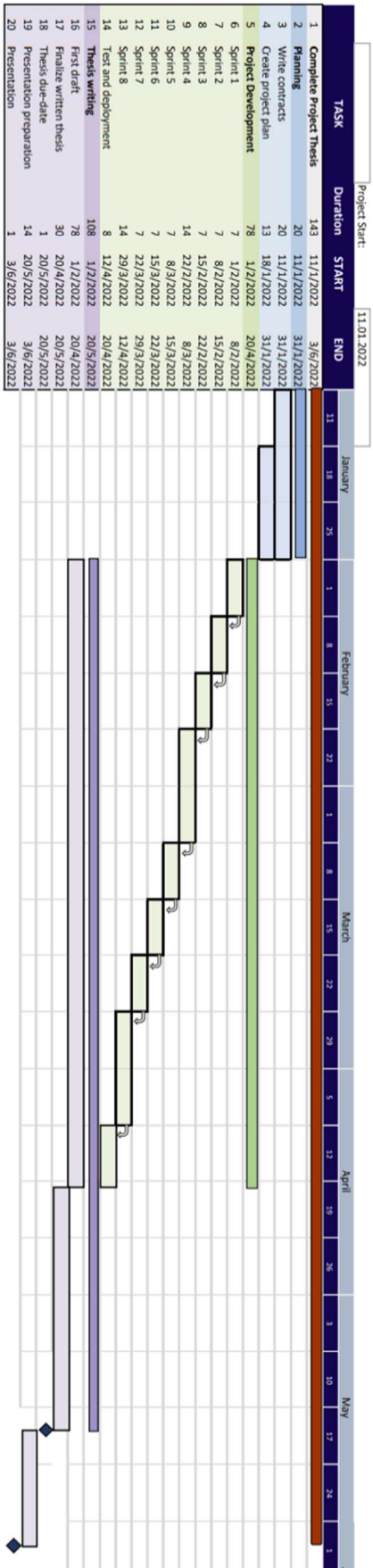
Essential tasks:

1. *Identify the most suitable tool and start designing the interface*
1. *Include the buttons that let navigate the file system and select input files and destination folder*
1. *Improve compatibility of deep-doLCE with the most common file formats used by film scanners*
1. *Offer to the user the possibility to select a Regio-Of-Interest (ROI)*
1. *Let the user choose the file format for the output, including a down sample option.*
1. *Run a test reconstruction on a single frame (a pre-flip left-right could be necessary)*
1. *Create the RUN button that start the reconstruction of the whole movie*
1. *Give the user feedback on the status of the process (i.e., output image, ETC, etc.)*

Additional tasks:

- *Expand the data augmentation with more transformations (e.g., blur, barrel/pincushion distortions)*
- *Allow for small curvature in the vectorization of the network's output*
- *Create a user guide to the application*
- *Add an option to review the software and give feedback.*

6.2 Gantt schema



	TASK	Duration	START	END
1	**Complete Project Thesis**	143	11/1/2022	3/6/2022
2	**Planning**	20	11/1/2022	31/1/2022
3	Write contracts	20	11/1/2022	31/1/2022
4	Create project plan	13	18/1/2022	31/1/2022
5	**Project Development**	78	1/2/2022	20/4/2022
6	Sprint 1	7	1/2/2022	8/2/2022
7	Sprint 2	7	8/2/2022	15/2/2022
8	Sprint 3	7	15/2/2022	22/2/2022
9	Sprint 4	14	22/2/2022	8/3/2022
10	Sprint 5	7	8/3/2022	15/3/2022
11	Sprint 6	7	15/3/2022	22/3/2022
12	Sprint 7	7	22/3/2022	29/3/2022
13	Sprint 8	14	29/3/2022	12/4/2022
14	Test and deployment	8	12/4/2022	20/4/2022
15	**Thesis writing**	108	1/2/2022	20/5/2022
16	First draft	78	1/2/2022	20/4/2022
17	Finalize written thesis	30	20/4/2022	20/5/2022
18	Thesis due-date	1	20/5/2022	20/5/2022
19	Presentation preparation	14	20/5/2022	3/6/2022
20	Presentation	1	3/6/2022	3/6/2022

References

[1] Dr. Giorgio Trumpy, and Dr. David Pfluger. “deep-doLCE: A New Machine Learning Approach for the Color Reconstruction of Digitized Lenticular Film” University of Zurich, 27 May 2021, https://www.film.uzh.ch/en/research/projects/verbund/deep-doLCE.html

[2] D’Aronco, Stefano, Giorgio Trumpy, David Pfluger, and Jan D. Wegner. “A Deep Learning Approach for Digital Color Reconstruction of Lenticular Films.” Transactions on Image Processing, no. under revision (n.d.).

Appendix B. Project Agreement

NTNU

Norwegian University of Science and Technology

Approved by the Pro-Rector for Education 10 December 2020

STANDARD AGREEMENT

on student works carried out in cooperation with an external organization

The agreement is mandatory for student works such as master's thesis, bachelor's thesis or project assignment (hereinafter referred to as works) at NTNU that are carried out in cooperation with an external organization.

Explanation of terms

Copyright

Is the right of the creator of a literary, scientific or artistic work to produce copies of the work and make it available to the public. A student thesis or paper is such a work.

Ownership of results

Means that whoever owns the results decides on these. The basic principle is that the student owns the results from their own student work. Students can also transfer their ownership to the external organization.

Right to use results

The owner of the results can give others a right to use the results – for example, the student gives NTNU and the external organization the right to use the results from the student work in their activities.

Project background

What the parties to the agreement bring with them into the project, that is what each party already owns or has rights to and which is used in the further development of the student's work. This may also be material to which third parties (who are not parties to the agreement) have rights.

Delayed publication (embargo)

Means that a work will not be available to the public until a certain period has passed; for example, publication will be delayed for three years. In this case, only the supervisor at NTNU, the examiners and the external organization will have access to the student work for the first three years after the student work has been submitted

1. Contracting parties

The Norwegian University of Science and Technology (NTNU) Department:
Supervisor at NTNU: email and telephone:
External organization: Contact person, email address and telephone number of the external organization: Giorgio Trumpy, giorgio.trumpy@ntnu.no, tlf: 916 74 430
Student: Qeis Khaled Qeis Date of birth: 27.07.1998
Other students, if applicable[1] Rúnar Ástvaldur Hedin 25.01.1991
Aleksander Kokowski 30.12.1997

The parties are responsible for clearing any intellectual property rights that the student, NTNU, the external organization or third party (which is not a party to the agreement) has to project background before use in connection with completion of the work. Ownership of project background must be set out in a separate annex to the agreement where this may be significant for the completion of the student work.

2. Execution of the work

The student is to complete: (Place an X)

A master's thesis	
A bachelor's thesis	X
A project assignment	
Another student work	

Start date: 02.02.2022
Completion date: 20.05.2022

The working title of the work is: GUI 4 deep-doLCE

[1] If several students co-author a work, they can all be listed here. The students then have joint rights to the work. If an external organization instead wants a separate agreement to be concluded with each student, this is done.

The responsible supervisor at NTNU has the overarching academic responsibility for the design and approval of the project description and the student's learning.

3. Duties of the external organization

The external organization must provide a contact person who has the necessary expertise to provide the student with adequate guidance in collaboration with the supervisor at NTNU. The external contact person is specified in Section 1.

The purpose of the work is to carry out a student assignment. The work is performed as part of the programme of study. The student must not receive a salary or similar remuneration from the external organization for the student work. Expenses related to carrying out the work must be covered by the external organization. Examples of relevant expenses include travel, materials for building prototypes, purchasing of samples, tests in a laboratory, chemicals. The student must obtain clearance for coverage of expenses with the external organization in advance.

The external organization must cover the following expenses for carrying out the work: NTNU will cover any cost that may become necessary

Coverage of expenses for purposes other than those listed here is to be decided by the external organization during the work process.

4. The student's rights

Students hold the copyright to their works [2]. All results of the work, created by the student alone through their own efforts, is owned by the student with the limitations that follow from sections 5, 6 and 7 below. The right of ownership to the results is to be transferred to the external organization if Section 5 b is checked or in cases as specified in Section 6 (transfer in connection with patentable inventions).

In accordance with the Copyright Act, students always retain the moral rights to their own literary, scientific or artistic work, that is, the right to claim authorship (the right of attribution) and the right to object to any distortion or modification of a work (the right of integrity).

A student has the right to enter into a separate agreement with NTNU on publication of their work in NTNU's institutional repository on the Internet (NTNU Open). The student also

[2] See Section 1 of the Norwegian Copyright Act of 15 June 2018 [Lov om opphavsrett til åndsverk]

has the right to publish the work or parts of it in other connections if no restrictions on the right to publish have been agreed on in this agreement; see Section 8.

5. Rights of the external organization

Where the work is based on or further develops materials and/or methods (project background) owned by the external organization, the project background is still owned by the external organization. If the student is to use results that include the external organization's project background, a prerequisite for this is that a separate agreement on this has been entered into between the student and the external organization.

Alternative a) (Place an X) General rule

X	The external organization is to have the right to use the results of the work

This means that the external organization must have the right to use the results of the work in its own activities. The right is non-exclusive.

Alternative B) (Place an X) Exception

	The external organization is to have the right of ownership to the results of the task and the student's contribution to the external organization's project

Justification of the external organization's need to have ownership of the results transferred to it:

6. Remuneration for patentable inventions

If the student, in connection with carrying out the work, has achieved a patentable invention, either alone or together with others, the external organization can claim transfer of the right to the invention to itself. A prerequisite for this is that exploitation of the invention falls within the external organization's sphere of activity. If so, the student is entitled to reasonable remuneration. The remuneration is to be determined in accordance with Section 7 of the Employees' Inventions Act. The provisions on deadlines in Section 7 apply correspondingly.

7. NTNU's rights

The submitted files of the work, together with appendices, which are necessary for assessment and archival at NTNU belong to NTNU. NTNU receives a right, free of charge, to use the results of the work, including appendices to this, and can use them for teaching and research purposes with any restrictions as set out in Section 8.

8. Delayed publication (embargo)

The general rule is that student works must be available to the public.

Place an X

X	The work is to be available to the public.

In special cases, the parties may agree that all or part of the work will be subject to delayed publication for a maximum of three years. If the work is exempted from publication, it will only be available to the student, external organization and supervisor during this period. The assessment committee will have access to the work in connection with assessment. The student, supervisor and examiners have a duty of confidentiality regarding content that is exempt from publication.

The work is to be subject to delayed publication for (place an X if this applies):

Place an X		Specify date
	one year	
	two years	
	three years	

The need for delayed publication is justified on the following basis:

If, after the work is complete, the parties agree that delayed publication is not necessary, this can be changed. If so, this must be agreed in writing.

Appendices to the student work can be exempted for more than three years at the request of the external organization. NTNU (through the department) and the student must accept this if the external organization has objective grounds for requesting that one or more appendices be exempted. The external organization must send the request before the work is delivered.

The parts of the work that are not subject to delayed publication can be published in NTNU's institutional repository – see the last paragraph of Section 4. Even if the work is subject to delayed publication, the external organization must establish a basis for the student to use all or part of the work in connection with job applications as well as continuation in a master's or doctoral thesis.

9. *General provisions*

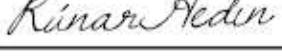
This agreement takes precedence over any other agreement(s) that have been or will be entered into by two of the parties mentioned above. If the student and the external organization are to enter into a confidentiality agreement regarding information of which the student becomes aware through the external organization, NTNU's standard template for confidentiality agreements can be used.

The external organization's own confidentiality agreement, or any confidentiality agreement that the external party has entered into in collaborative projects, can also be used provided that it does not include points in conflict with this agreement (on rights, publication, etc). However, if it emerges that there is a conflict, NTNU's standard contract on carrying out a student work must take precedence. Any agreement on confidentiality must be attached to this agreement.

Should there be any dispute relating to this agreement, efforts must be made to resolve this by negotiations. If this does not lead to a solution, the parties agree to resolution of the dispute by arbitration in accordance with Norwegian law. Any such dispute is to be decided by the chief judge (sorenskriver) at the Sør-Trøndelag District Court or whoever he/she appoints.

This agreement is signed in four copies, where each party to this agreement is to keep one copy. The agreement comes into effect when it has been signed by NTNU, represented by the Head of Department.

Signatures:

Head of Department: Date:	
Supervisor at NTNU: Date:	
External organization: Date: 28/01/2022	
Student: Date: 29.01.2022	
Other students, if applicable	
	

NTNU

Norwegian University of Science and Technology

Approved by the Pro-Rector for Education 10 December 2020

Acceptance – transfer of rights to NTNU

This acceptance form must be used when students are to transfer IP rights to NTNU. Intellectual property rights (IPR) is a collective term for copyright, patent rights, design rights, and trademark rights, among others.

The same form is used in connection with bachelor's and master's thesis or other student works.

The point of departure is that students themselves own the IPR they create as part of their studies at NTNU, but other terms can be agreed on.

Students will typically need to accept a transfer of IPR if they participate in EU projects or projects that are funded by the Research Council of Norway. Both the EU and the Research Council require participants in projects to transfer intellectual property rights to the results to the organization that has received funding. This is to ensure that it will be possible in practice to take advantage of the project results so that they can benefit society. The same need may also arise in other projects.

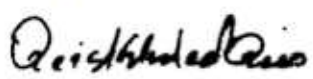
Even if ownership rights are transferred to NTNU, students always retain the moral rights to their own literary, scientific or artistic work, that is, the right to claim authorship (the right of attribution) and the right to object to any distortion or modification of a work (the right of integrity). This follows from the Norwegian Copyright Act of 15 June 2018. Section 5. This means that the student is entitled to attribution if for example, a bachelor's or master's thesis is used.

Transfer of rights to NTNU applies to the following project and is justified by the following:

I, student .....Qeis Khaled Qeis..... (block capitals), date of birth.....27.07.1998

accept that IP rights are transferred to NTNU.

Signature

Student:

Date: 29.01.2022

NTNU

Norwegian University of Science and Technology

Approved by the Pro-Rector for Education 10 December 2020

Acceptance – transfer of rights to NTNU

This acceptance form must be used when students are to transfer IP rights to NTNU. Intellectual property rights (IPR) is a collective term for copyright, patent rights, design rights, and trademark rights, among others.

The same form is used in connection with bachelor's and master's thesis or other student works.

The point of departure is that students themselves own the IPR they create as part of their studies at NTNU, but other terms can be agreed on.

Students will typically need to accept a transfer of IPR if they participate in EU projects or projects that are funded by the Research Council of Norway. Both the EU and the Research Council require participants in projects to transfer intellectual property rights to the results to the organization that has received funding. This is to ensure that it will be possible in practice to take advantage of the project results so that they can benefit society. The same need may also arise in other projects.

Even if ownership rights are transferred to NTNU, students always retain the moral rights to their own literary, scientific or artistic work, that is, the right to claim authorship (the right of attribution) and the right to object to any distortion or modification of a work (the right of integrity). This follows from the Norwegian Copyright Act of 15 June 2018. Section 5. This means that the student is entitled to attribution if for example, a bachelor's or master's thesis is used.

Transfer of rights to NTNU applies to the following project and is justified by the following:

I, student ALEKSANDER KOKOWSKI..... (block capitals), date of birth 30.12.97.....

accept that IP rights are transferred to NTNU.

Signature

Student: Aleksander Kokowski Date: 30.01.22

NTNU

Norwegian University of Science and Technology

Approved by the Pro-Rector for Education 10 December 2020

Acceptance – transfer of rights to NTNU

This acceptance form must be used when students are to transfer IP rights to NTNU. Intellectual property rights (IPR) is a collective term for copyright, patent rights, design rights, and trademark rights, among others.

The same form is used in connection with bachelor's and master's thesis or other student works.

The point of departure is that students themselves own the IPR they create as part of their studies at NTNU, but other terms can be agreed on.

Students will typically need to accept a transfer of IPR if they participate in EU projects or projects that are funded by the Research Council of Norway. Both the EU and the Research Council require participants in projects to transfer intellectual property rights to the results to the organization that has received funding. This is to ensure that it will be possible in practice to take advantage of the project results so that they can benefit society. The same need may also arise in other projects.

Even if ownership rights are transferred to NTNU, students always retain the moral rights to their own literary, scientific or artistic work, that is, the right to claim authorship (the right of attribution) and the right to object to any distortion or modification of a work (the right of integrity). This follows from the Norwegian Copyright Act of 15 June 2018. Section 5. This means that the student is entitled to attribution if for example, a bachelor's or master's thesis is used.

Transfer of rights to NTNU applies to the following project and is justified by the following:

I, student RÚNAR ÁSTVALDUR HEDIN (block capitals), date of birth 25.01.1991.

accept that IP rights are transferred to NTNU.

Signature

Student: Rúnar Hedin Date: 29.01.2022

NTNU

Norwegian University of Science and Technology

Approved by the Pro-Rector for Education 10 December 2020

STANDARD template between a student and an external organization for student work such as master's thesis or another student work in cooperation with an external organization, cf. Clause 9 in the standard agreement on student work carried out in cooperation with an external organization.

Student at NTNU: Qeis Khaled Qeis Date of birth: 27.07.1998		
Other students, if applicable	Runar Astvaldur Hedin 25.01.1991	Aleksander Kokowski 30.12.1997
External organization: Giorgio Trumpy - NTNU		

1. The student is to carry out work in cooperation with an external organization that is part of his/her course of study at NTNU.

2. The student undertakes to maintain secrecy regarding what he/she learns about technical equipment, procedures as well as operational and business matters that for competitive reasons have importance for the external organization. It is the responsibility of the external organization to make it absolutely clear what this information includes.

3. The student is obliged to maintain secrecy about this for 5 years from the date he/she completed work at the organization.

4. The confidentiality requirement does not apply to information that:
a) was in the public domain when it was received
b) was lawfully received from a third party without any agreement concerning secrecy
c) was developed by the student independently of information received
d) the parties are obliged to provide in accordance with law or regulations or by order of a public authority.

Signaturs

Student: Date: 29.01.2022	Qeis Khaled Qeis
Other students, if applicable:	Runar Hedin Aleksander Kokowski
External organization: Date: 28/01/2022	

Contract for GUI 4 deep-doLCE

Group members: Aleksander Kokowski, Runar Astvaldur Hedin, Qeis Khaled Qeis

.....

Goals

Effect goals

We aim to achieve a grade of A, but will also keep our expectations a bit lower but we will try our best regardless.

Get to know each other, build up trust and respect for each other to increase the motivation for the thesis

We will have to be honest with each other and communicate as much as possible whenever there might be an issue.

Able to work together effectively

To achieve this, we will set regular meetings every week, while remaining flexible regarding work schedules etc. More meetings can be set up when/if needed.

Work flexibility and solution-oriented behaviour

Meetings should be scheduled in advance at a time that everyone agrees on and fits everyone's schedule. Due to work schedule, these meetings will be held online as much as possible.

End result goal

Deliver all of the parts in time

Aim to always start our work as early as possible and give ourselves plenty of room before the deadline to fix everything that might need fixing.

Achieve a chosen grade

To achieve this, we will work as hard as we can, be innovative, and creative.

Roles and dividing workload

What roles do we need – what kind of structure should we have?

Team leadership

Organizing meetings, replying to emails from clients and the university staff. Qeis Khaled Qeis will be responsible for that role.

Archiver/recordkeeper

Takes notes during meetings about what was discussed, also notes when, where, how long each meeting is, and who was in the meeting. Runar Astvaldur Hedin will be responsible for that role.

Responsible for submission and quality assurance

Proofreads and submits each document/thesis (QA for code will be a shared responsibility for the entire group). Aleksander Kokowski will be responsible for that role.

Procedures

Convening meetings

We will have a scheduled meeting between the group and the university staff every monday at around 10am. These meetings will be held on either Zoom or Microsoft Teams.

Notification about absence or other related issues.

Everyone is required to notify the others in case of absence or running late.

Document management

Thesis will be written through Microsoft Teams or Google Docs.

Interaction

Presence and commitment

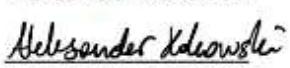
Everyone is expected to focus on the ongoing work while we're working together.

Disagreement, breach of contract

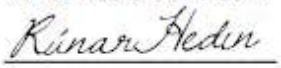
If anyone does not show up regularly in meetings or fails to do their tasks, then that person will be talked to, and the group will collectively try to solve the problem. If the problem persists then the university staff might be contacted.

Signed,

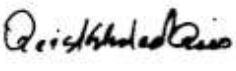
Aleksander Kokowski



Runar Astvaldur Hedin



Qeis Khaled Qeis



Appendix C. Client and supervisor meetings

Participants	Date	Start Tim	End time	Hours	Minutes	Notes
Giorgio	02.12.2022	11:00	11:20	0	20	Introductions and getting an idea of what the project/thesis is about
Supervisors	10.01.2022	10:00	10:15	0	15	Introductions
Giorgio	10.01.2022	11:00	11:10	0	10	Update on what we've done so far etc
Giorgio	11.01.2022	13:30	14:20	0	50	Detailed information about what Giorgio and his team's code does. Getting an idea of the tasks we need to do for out part.
Supervisors	17.01.2022	09:30	09:45	0	15	Update on what we've done so far and getting ideas about what we're supposed to do regarding the contract and plan.
Group	23.01.2022	12:55	14:15	1	20	Finished the first draft of both the contracts and plan.
Supervisors	24.01.2022	10:00	10:15	0	15	Got Sony's input on our contract and plan
Group	25.01.2022	12:00	13:10	1	10	Worked on the plan and found the right contract to sign together with the company
Supervisors	27.01.2022	14:15	14:45	0	30	Got Sony's feedback on the plan so far and where to go from there
Giorgio	28.01.2022	12:00	12:50	0	50	Questions and feedback regarding the planning. Went over contracts as well.
Supervisors	31.01.2022	10:00	10:15	0	15	Steven checked out the finished draft of the plan contracts and we got the go-ahead to deliver.
Group	02.02.2022	11:00	11:20	0	20	Sprint1 meeting, got every program we'd need for this sprint up and running and officially started the project
Supervisors	07.02.2022	10:00	10:05	0	5	Discussed what we did the previous week, which was to create a wireframe, make accounts on the different tools we're planning on using, and read up & chose the python GUI tool kivy
Supervisors	14.02.2022	10:00	10:10	0	10	Weekly status update
Supervisors	21.02.2022	10:00	10:15	0	15	Showed discussed the GUI we've made so far and got some feedback from that.
Giorgio	21.02.2022	11:00	11:35	0	35	Discussed input and output format of deepdoLCE. Talked a bit about the displayed preview images of a movie, image flipping and a popup window to flip those preview images before continuing with the render.

supervisors	28.02.2022	10:00	10:25	0	25	Feedback on buttons and implimentation of buttons that run the deep-doLCE code. We need to optimize the code on our end to make the program run faster since currently it takes roughly 10 seconds to colorize a 30MB image file.

Supervisors	07.03.2022	13:00	13:15	0	15	Weekly status update. Discussed issues with last weeks implimentation ideas, more specifically about using a python plugin to process images faster and the need to properly understand that plugin.
Supervisors	14.03.2022	10:00	10:05	0	5	Weekly status meeting. ROI button implementation, and other button functionality.
Supervisors	21.03.2022	10:00	10:05	0	5	Weekly status meeting. Help button, flip button and running the software on the GPU rather than CPU amongst other things.
Giorgio	25.03.2022	15:00	15:30	0	30	Got information about where we can find the specific functions in their code and also talked and confirmed the buttons he wanted.
Supervisors	28.03.2022	10:00	10:10	0	10	Weekly status meeting. Talked about being finished with the project relatively early and concerns the supervisors had with that.
Giorgio	01.04.2022	09:15	10:20	1	5	Clarification on deep-doLCE code and got some help with optimizing code.
Supervisors	04.04.2022	10:00	10:20	0	20	Discussed our individual contributions to the project so far and started laying out our plan for the next couple of weeks. Discussed how we're going to achieve our desired grade as well.
Giorgio	07.04.2022	11:30	12:10	0	40	Showed Giorgio the current progress. Worked with him to optimize the way films/images are temporarily stored while they're being processed (ram vs disk storage).
Supervisors	11.04.2022	10:00	10:05	0	5	Short meeting regarding last week and our plans for continued work this week.
Giorgio	19.04.2022	10:00	10:30	0	30	Met with Giorgio for feedback and help with the backend coding
Giorgio	22.04.2022	14:30	15:30	1	0	Met with Giorgio for feedback and help with the backend coding
Supervisors	25.04.2022	10:00	10:20	0	20	Showed the supervisors our running GUI and briefly our thesis so far.

Supervisors+Giorgio	03.05.2022	13:00	14:15	1	15	Went through the whole app in detail in a physical meeting with the supervisors and Giorgio. Got feedback from them as well.
Supervisors	09.05.2022	10:00	10:30	0	30	Went through the app again with the supervisors and showed them our final results.
Usability tests	10.05.2022	10:30	12:15	1	45	Got user feedback from two film archivists.
Usability tests	10.05.2022	17:30	18:30	1	0	Got user feedback from film archivist.
Supervisors	16.05.2022	10:00	10:15	0	15	Discussed the next couple of days and got a few tips regarding the thesis
Total:	////////////	//////////	//////////	14	50	14:50 total