

Oslomet

# FORPROSJEKTRAPPORT

- Hovedprosjekt i data og informasjonsteknologi
  - Magnifier-Wild
    - Gruppe 51.

03.01.2023

## Innhold:

1.Presentasjon .....	2
1.1 Gruppemedlemmer: .....	2
1.2 Oppdragsgiver .....	2
1.3 Veileder .....	2
1.4 Veileder ved OsloMet .....	2
1.5 Oppgaven .....	3
Sammendrag .....	3
Dagens situasjon.....	3
Mål og rammebetingelser .....	3
4.2 Oppdragsgivers mål/ Kravspesifikasjon .....	4
4.5 Programmeringsspråk og rammeverk.....	4
Løsninger/alternativer.....	5
Analyse av virkninger .....	6

# 1.Presentasjon

## 1.1 Gruppemedlemmer:

Navn	Studieretning	E-post
Nechrvan Murad	Dataingeniør	s351927@oslomet.no
Mohammad Khaled	Dataingeniør	s351903@oslomet.no
Fadl El Nassani	Dataingeniør	S349427@oslomet.no

## 1.2 Oppdragsgiver

Frode eika sandens

Telfon: +47 67 23 50 03

E-post: [frode-eika.sandnes@oslomet.no](mailto:frode-eika.sandnes@oslomet.no)

Adresse: Pilestredet 35, 0166 Oslo, Kontor nummer: PS331

Forsker på brukergrensesnitt, opptatt av innovative interaksjonsformer og gjenbruk av ferdigheter samt universell utforming. Arbeider med mønstergjenkjenning, bildeanalyse og intelligente systemer. Faglærer for faget Menneske maskin interaksjon.

## 1.3 Veileder

Navn	Stilling	Epost
Frode eika sandens	professor	<a href="mailto:frode-eika.sandnes@oslomet.no">frode-eika.sandnes@oslomet.no</a>

## 1.4 Veileder ved OsloMet

Navn	Stilling	Epost
Frode eika sandens	professor	<a href="mailto:frode-eika.sandnes@oslomet.no">frode-eika.sandnes@oslomet.no</a>

## 1.5 Oppgaven

Problemstillingen:

Mange brukere tar bilder av liten tekst i omgivelsene som kan være vanskelig å lese (for eksempel skilt og informasjonstavler ((og bruker zoom til å lese detaljer. Problemet med dette er at brukeren må panorere i to-dimensjoner (horisontalt og vertikalt).

## Sammendrag

Dette prosjektet går ut på å implementere en forstørrelsesapp for nettlesere for mobile enheter som smarttelefoner og nettbrett. Appen skal la brukeren ta et bilde av tekst – for eksempel en informasjonstavle, og deretter utføre perspektivkorleksjon og dele opp bildet for relflow (reformatering) av innholdet. Målet med dette prosjektet er å implementere en løsning på dette problemet med en intelligent forstørrelsesapp.

## Dagens situasjon

Mange bruker smart telefoner i den tiden, for å gjøre dagens i dag lettere og. Å lage en Magnifier for brukere vil gjøre det mer effektivt og oversiktlig for gamle mennesker og de som har dårlig syn. Den skal i tillegg kunne ivareta medlemmenes personvern og sikkerhet.

## Mål og rammebetingelser

Det er rett og slett å lage et program som kan gjøre noe annerledes og samtidig bra for menneskelighet. Mange trenger hjelp i den moderne verden, der mange bruker de smarte funksjonene og enheter og disse enhetene fører av og til dårlig syn og synsproblemer.

### 4.1 Mål

- Målet med dette prosjektet er å implementere en løsning på dette problemet med en intelligent forstørrelsesapp.
- En Bruker skal ta bildet av tavle, papir eller informasjonstavle med moblie og deretter utføre perspektivkorleksjon og dele opp bildet for relflow (reformatering) av innholdet.

- Lære om de forskjellige tekniker og funksjoner i OpenCV.JS biblioteket.
- Open Camorra fra mobil eller datamaskin og ta et bilde av tavle eller papir deretter rette bildet hvis det er skeiv og trekke ut teksten i bildet som array av små bilder og reflow

## 4.2 Oppdragsgivers mål/ Kravspesifikasjon

### 4.3 Rammebetingelser

- Gruppen skal ha møte med intern veileder hver fredag, og møte med oppdragsgiver etter hver fullført sprint.
- Gruppen skal til enhver tid følge punktene som står skrevet i den signerte teamkontrakten
- Prosjektet skal leveres Ca. i 26.05.2021 til OsloMet
- Prosjektet skal dokumenteres underveis i prosessen

### 4.4 Verktøy

- IntelliJ IDEA/ visual studio code for programmering
- GitHub for versjonskontroll-/håndtering
- Figma, Sketch.
- Google Drive for lagring av dokumenter, filer og notater som alle i gruppen har tilgang til å lese og endre som for eksempel prosjektdagboken og sluttrapport
- Discord for kommunikasjon og samarbeid i gruppen via blant annet chat og møter
- Microsoft Teams for kommunikasjon og samarbeid i gruppen og med oppdragsgiver via chat og møter

## 4.5 Programmeringsspråk og rammeverk

Vanilla JavaScript (noe CSS, HTML).

Mulige tredjeparts biblioteker: OpenCV.js

o Browser/nettleser

o Media/camera API

o Digital image processing

o Reflow engine

o Mobile user interface

## Løsninger/alternativer

	Løsning	Fordeler	Ulemper
1	<ul style="list-style-type: none"><li>• HTML, CSS</li><li>• JavaScript</li><li>• OpenCV.js</li><li>• Eventuell tesseract-ocr.js og andre biblioteker for bilder behandling</li></ul>	<p>Et sterkt programmeringsspråk for web applikasjon.</p> <p>Javascript har flere biblioteker som gjelder bildebehandling.</p> <p>Har alle grunnleggende for å bygge et program.</p> <p>Alle i gruppen har erfaring med teknologiene fra tidligere prosjekter.</p> <p>JavaScript trenger noen rammeverk til å å kjøre koden</p>	<p>Er mer krevende, ikke lett som andre programmeringsspråk</p> <p>FKs: Python.</p> <p>Man må skrive lange koder for å oppnå målet.</p>
2	<ul style="list-style-type: none"><li>• Python</li><li>• NumPy</li></ul>	<p>OpenCV er Et fint bibliotek med flere funksjoner for å behandle bilde og video</p> <p>Biblioteket er passende med Python.</p> <p>Man kan bruke Numpy med openCv.js.</p> <p>Numpy-matriser tar mindre minneplass og gir bedre kjørehastighet sammenlignet med lignende datastrukturer i Python.</p>	<p>Det er vanskelig å få opp resultatene på HTML som kreves</p> <p>mere biblioteker som Flask eller Streamlit for å vise resultater in html</p>
3	<ul style="list-style-type: none"><li>• C++</li><li>• OpenCV</li></ul>	<p>Fleksibel, Enkelt å finne feil, raskt fordi C++ får tilgang over hardware.</p>	<p>C++ er mer rettet mot PC applikasjoner, men</p>

		De fleste nye OpenCV algoritmer blir skrevet på C++	vi skal lage en webapplikasjon. C++ er rettet mot tunge programmer og litt vanskelig å implementere biblioteker.
--	--	---	--

## Analyse av virkninger

### Alternativ 1:

Javascript er enkelt og passende med webapplikasjoner og har flere biblioteker som kan spare tid og kostander. I tillegg er det enkelt å vise resultatene in HTML side eller kalle på forskjellige funksjoner fra HTML (GUI).

OpenCV.js et kraftig JavaScript bibliotek som brukes til bildebehandling og bildegjenkjenning. Vi kan eventuelt bruke andre biblioteker som tesseract-ocr.js for å kunne trekke ut teksten fra et bilde. Dette skaper et datamaskin syn.

OpenCV.js integrerer godt med JavaScript uten noen problem. I tillegg er det flere ressurser som finnes in openCV sin web side for å lære de forskjellige funksjoner for bilde og video behandling og hva slags matematikk står bak de forskjellige funksjonene.

### Alternative 2:

Python med Opencv.js og Numpy bibliotek er fint alternativ, men de er kregende å vise resultatene in HTML og trenger ekstra biblioteker.

Vi kan oppleve problemer når vi sammenligner verdier i Python-tolken. I tillegg krever den en sammenhengende tildeling av minne som Innsettings- og slettingsoperasjoner blir kostbare ettersom data lagres i sammenhengende minneplasseringer. I tillegg er Python Svak i databehandling og nettlesere som kan føre til Sikkerhets gap.<sup>1</sup>

### Alternative 3:

---

<sup>1</sup> <https://data-flair.training/blogs/advantages-and-disadvantages-of-python/>

C++ kan brukes mest med datamaskin applikasjoner og ikke enkelt som javascript eller Python. I tillegg har vi ikke nok kompetanse til å bruke C++ språk som krever mye forgåelse av oppbygning av algoritmer.

OpenCV (C++) har et lite sett med maskinlæringsalgoritmer sammenlignet med valgene. I tillegg er det vanskelig med visualisering og Debugging særlig når du skriver ny algoritme.<sup>2</sup>

---

<sup>2</sup> <https://learnopencv.com/opencv-c-vs-python-vs-matlab-for-computer-vision/>