Table of Contents

# Medical Text Extractor - Brukerdokumentasjon

## Innholdsfortegnelse

1. [Introduksjon](#introduksjon)
2. [Installasjonsveiledning](#installasjonsveiledning)
3. [Komme i gang](#komme-i-gang)
4. [Funksjonalitet](#funksjonalitet)
5. [Systemkrav](#systemkrav)
6. [Brukerveiledning](#brukerveiledning)
7. [Feilsøking](#feilsøking)
8. [API-dokumentasjon](#api-dokumentasjon)
9. [Kontaktinformasjon](#kontaktinformasjon)

## Introduksjon

Medical Text Extractor er et verktøy utviklet for å hjelpe helsepersonell med å raskt hente ut administrasjonsinstruksjoner for medisiner fra bilder. Applikasjonen overvåker utklippstavlen for bilder, bruker OCR (Optical Character Recognition) for å trekke ut tekst, identifiserer medisinnavn og viser administrasjonsinstruksjoner automatisk.

### Hovedfunksjoner

* **Automatisk overvåking av utklippstavlen**: Oppdager automatisk når bilder kopieres til utklippstavlen
* **OCR-behandling**: Trekker ut tekst fra bilder ved hjelp av Tesseract OCR
* **Tekstanalyse**: Identifiserer norske medisinske termer som “rekvirent” og “Legemiddel”
* **Databaseintegrasjon**: Lagrer og henter administrasjonsinstruksjoner for medisiner
* **Systemstatusikon**: Gir enkel tilgang til å starte/stoppe overvåkingen og avslutte applikasjonen
* **Brukernotifikasjoner**: Viser dialogbokser med administrasjonsinstruksjoner når medisiner blir identifisert

## Installasjonsveiledning

### Forutsetninger

Før du installerer Medical Text Extractor, må du ha følgende programvare installert:

1. **Python 3.9**: Kan lastes ned fra [python.org](https://www.python.org/downloads/)
2. **Conda**: Anbefalt for å håndtere virtuelle miljøer, kan lastes ned fra [conda.io](https://docs.conda.io/en/latest/miniconda.html)
3. **Tesseract OCR**: Nødvendig for OCR-funksjonalitet, kan lastes ned fra [UB-Mannheim](https://github.com/UB-Mannheim/tesseract/wiki)

### Installasjonstrinn

1. **Klon eller last ned prosjektet**:

* git clone https://github.com/tskaret/MedicalTextExtractor.git  
  cd MedicalTextExtractor
* **Merk:** Du vil bli bedt om å oppgi passord for å klone repositoriet.

1. **Opprett Conda-miljø**:

* powershell -ExecutionPolicy Bypass -File setup\_conda\_env.ps1
* Dette vil opprette et Conda-miljø med navnet med\_text\_env i d:\Conda.env\med\_text\_env.

1. **Aktiver Conda-miljøet**:

* .\activate\_env.bat

1. **Verifiser installasjonen**:

* python -c "import pytesseract; print('Tesseract er tilgjengelig')"
* Hvis du ikke ser noen feilmeldinger, er installasjonen vellykket.

## Komme i gang

### Starte applikasjonen

For å starte Medical Text Extractor med systemstatusikon:

1. Aktiver Conda-miljøet:

* .\activate\_env.bat

1. Start den forbedrede utklippstavleovervåkingen:

* python start\_enhanced\_clipboard\_service.py

1. Et systemstatusikon vil vises i systemstatusfeltet. Høyreklikk på ikonet for å se alternativer:
   * **Start overvåking**: Starter overvåking av utklippstavlen
   * **Stopp overvåking**: Stopper overvåking av utklippstavlen
   * **Avslutt**: Avslutter applikasjonen

### Bruke applikasjonen

1. Trykk tasten “Prt Scrn” (Print Screen) for å ta bilde av skjermen og aktivere programmet. Dersom programmet oppdager legemiddel i bildet av Alfa-skjermen, vil det kopiere bruksanvisningen (administreringen). Den kan kopieres ut ved å trykke Ctrl+V eller høyre knapp på musen og velge “Lim inn” i Alfa.
2. Alternativt kan du kopiere et bilde som inneholder medisinsk informasjon til utklippstavlen (f.eks. ved å ta et skjermbilde eller kopiere et bilde fra en annen applikasjon)
3. Applikasjonen vil automatisk oppdage bildet, trekke ut tekst og analysere den
4. Hvis medisininformasjon blir funnet, vil en dialogboks vise administrasjonsinstruksjonene
5. Administrasjonsinstruksjonene blir også kopiert til utklippstavlen for enkel liming inn i andre applikasjoner

## Funksjonalitet

### Utklippstavleovervåking

Applikasjonen overvåker kontinuerlig utklippstavlen for nye bilder. Når et bilde oppdages, blir det behandlet for å trekke ut medisinsk informasjon.

### OCR-behandling

Optical Character Recognition (OCR) brukes for å trekke ut tekst fra bilder. Applikasjonen bruker Tesseract OCR, som er en kraftig OCR-motor som støtter flere språk, inkludert norsk.

### Tekstanalyse

Den ekstraherte teksten analyseres for å identifisere viktig medisinsk informasjon, inkludert: - HPR-nummer (Helsepersonellregisternummer) - Rekvirent (lege eller annet helsepersonell) - Legemiddelnavn

### Databaseintegrasjon

Applikasjonen lagrer og henter administrasjonsinstruksjoner for medisiner fra en lokal database. Dette gjør det mulig å raskt hente instruksjoner for kjente medisiner.

### Systemstatusikon

Et systemstatusikon gir enkel tilgang til applikasjonens funksjoner, inkludert: - Starte/stoppe overvåking av utklippstavlen - Avslutte applikasjonen

### Brukernotifikasjoner

Applikasjonen viser dialogbokser med administrasjonsinstruksjoner når medisiner blir identifisert. Dette gir umiddelbar tilgang til viktig informasjon.

## Systemkrav

### Maskinvarekrav

* **Prosessor**: Minimum 2 GHz dual-core
* **Minne**: Minimum 4 GB RAM
* **Lagring**: Minimum 500 MB ledig diskplass

### Programvarekrav

* **Operativsystem**: Windows 10 eller nyere
* **Python**: Version 3.9
* **Conda**: For håndtering av virtuelle miljøer
* **Tesseract OCR**: For OCR-funksjonalitet

### Avhengigheter

Følgende Python-pakker er nødvendige: - opencv-python - for bildebehandling - pillow - for bildemanipulering - pyautogui - for skjermbildedeteksjon og automatisering - pynput - for tastatur-/museovervåking - pyperclip - for utklippstavleoperasjoner - pytesseract - Python-wrapper for Tesseract OCR - win10toast - for Windows-varsler - pystray - for systemstatusikon - pywin32 - for Windows API-tilgang - psutil - for prosessovervåking

## Brukerveiledning

### Overvåking av utklippstavlen

1. **Start overvåking**:
   * Høyreklikk på systemstatusikonet
   * Velg “Start overvåking”
2. **Stopp overvåking**:
   * Høyreklikk på systemstatusikonet
   * Velg “Stopp overvåking”

### Behandle bilder manuelt

Hvis du ønsker å behandle et bilde manuelt:

1. Kopier bildet til utklippstavlen
2. Vent på at applikasjonen behandler bildet
3. En dialogboks vil vise administrasjonsinstruksjonene hvis en medisin blir identifisert

### Legge til nye medisiner i databasen

For å legge til en ny medisin i databasen:

1. Aktiver Conda-miljøet:

* .\activate\_env.bat

1. Kjør skriptet for å legge til medisiner:

* python add\_medication\_improved.py

1. Følg instruksjonene for å legge til medisininformasjon:
   * Medisinnavn
   * Administrasjonsinstruksjoner
   * Andre relevante opplysninger

### Vise lagrede medisiner

For å vise alle medisiner i databasen:

1. Aktiver Conda-miljøet:

* .\activate\_env.bat

1. Kjør skriptet for å vise medisiner:

* python view\_medications.py

## Feilsøking

### Vanlige problemer

#### Applikasjonen starter ikke

**Problem**: Applikasjonen starter ikke når du kjører start\_enhanced\_clipboard\_service.py.

**Løsning**: 1. Sjekk at Conda-miljøet er aktivert: powershell .\activate\_env.bat 2. Sjekk at alle nødvendige pakker er installert: powershell pip list 3. Sjekk loggfilen service\_starter.log for feilmeldinger: powershell Get-Content -Path service\_starter.log -Tail 20

#### OCR fungerer ikke som forventet

**Problem**: Applikasjonen oppdager bilder, men trekker ikke ut tekst korrekt.

**Løsning**: 1. Sjekk at Tesseract OCR er installert og konfigurert korrekt 2. Sjekk at bildet har god kvalitet og er lesbart 3. Prøv å behandle bildet manuelt med debug-skriptet: powershell python debug\_clipboard\_monitor.py 4. Sjekk loggfilen clipboard\_monitor.log for feilmeldinger: powershell Get-Content -Path clipboard\_monitor.log -Tail 20

#### Applikasjonen finner ikke medisiner

**Problem**: Applikasjonen trekker ut tekst, men finner ikke medisininformasjon.

**Løsning**: 1. Sjekk at teksten inneholder medisinnavn i et format som applikasjonen kan gjenkjenne 2. Sjekk at medisinen finnes i databasen: powershell python lookup\_medication.py [medisinnavn] 3. Prøv å legge til medisinen manuelt: powershell python add\_medication\_improved.py

### Loggfiler

Applikasjonen genererer flere loggfiler som kan være nyttige for feilsøking:

* **clipboard\_monitor.log**: Loggfil for utklippstavleovervåkingen
* **service\_starter.log**: Loggfil for oppstartsprosessen
* **extracted\_text.txt**: Inneholder tekst som er trukket ut fra det siste behandlede bildet
* **clipboard\_results.txt**: Inneholder resultater fra den siste behandlingen

## API-dokumentasjon

### Hovedkomponenter

#### EnhancedClipboardMonitor

class EnhancedClipboardMonitor:  
 """  
 Overvåker utklippstavlen for nye bilder, trekker ut medisininformasjon,  
 viser administrasjonsinstruksjoner og gir et systemstatusikon for brukerkontroll.  
 """  
   
 def \_\_init\_\_(self):  
 """Initialiser utklippstavleovervåkeren."""  
   
 def start(self):  
 """Start overvåking av utklippstavlen."""  
   
 def stop(self):  
 """Stopp overvåking av utklippstavlen."""  
   
 def run(self):  
 """Kjør den forbedrede utklippstavleovervåkeren med systemstatusikon."""

#### OCRProcessor

class OCRProcessor:  
 """  
 Behandler bilder med OCR for å trekke ut tekst.  
 """  
   
 def \_\_init\_\_(self):  
 """Initialiser OCR-prosessoren."""  
   
 def extract\_text(self, image):  
 """Trekk ut tekst fra et bilde ved hjelp av OCR."""

#### ImprovedTextAnalyzer

class ImprovedTextAnalyzer:  
 """  
 Analyserer tekst for å identifisere medisininformasjon.  
 """  
   
 def \_\_init\_\_(self):  
 """Initialiser tekstanalysatoren."""  
   
 def analyze\_text(self, text):  
 """Analyser tekst for å identifisere medisininformasjon."""

### Hjelpefunksjoner

#### lookup\_medication

def lookup\_medication(medication\_name, silent=False):  
 """  
 Slå opp en medisin i databasen.  
   
 Args:  
 medication\_name (str): Navnet på medisinen å slå opp  
 silent (bool): Hvis True, skriv ikke ut resultater til konsollen  
   
 Returns:  
 list: Liste over medisiner som matcher søket  
 """

## Kontaktinformasjon

For spørsmål, tilbakemeldinger eller støtte, vennligst kontakt:

* **GitHub**: github.com/tskaret/MedicalTextExtractor