

Poradnik: Aplikacje GUI do Trenowania i Predykcji

1 Wprowadzenie

Dokument opisuje dwie aplikacje GUI stworzone w języku Python. Aplikacje te służą do trenowania modeli oraz inferencji. Obie aplikacje zostały zaprojektowane w celu umożliwienia użytkownikowi łatwej konfiguracji parametrów trenowania oraz zastosowania modeli do przewidywań na podstawie obrazów.

2 Aplikacja 1: `gui.py`

Aplikacja `gui.py` oferuje interfejs graficzny, który umożliwia użytkownikowi wybór między trybem trenowania modelu a trybem inferencji. W każdym z tych trybów dostępne są różne parametry, które użytkownik może konfigurować, aby dostosować działanie aplikacji do swoich potrzeb.

2.1 Tryb trenowania

W trybie trenowania, użytkownik ma możliwość wprowadzenia parametrów, takich jak liczba epok, wielkość partii (batch size), współczynnik uczenia się (learning rate), a także wybór modelu, funkcji straty, optymalizatora i innych parametrów. Wyboru dokonuje się poprzez listy rozwijane oraz pola tekstowe.

The image shows a software window titled "Training and Inference Application". It has two tabs: "Training" and "Inference". The "Training" tab is active. The form contains the following fields and controls:

- Number of epochs: 300
- Batch size: 6
- Learning rate: 1e-3
- Annotator: 2 (dropdown)
- Model: smpUNet++ (dropdown)
- Loss function: CrossEntropyLoss (dropdown)
- Optimizer: Adam (dropdown)
- Scheduler: CosineAnnealingLR (dropdown)
- Mode: intersection (dropdown)
- Place: lab (dropdown)
- Augmentation Type: ClassSpecificAugmentation (dropdown)
- K parameter: 5
- ☐ Use Augmentation
- Start Training button

Figure 1: Widok pełnego GUI dla trybu trenowania.

2.2 Tryb inferencji

Tryb inferencji pozwala użytkownikowi na wczytanie wytrenowanego modelu oraz wykonanie predykcji na nowych danych. Formularz trybu predykcji zawiera m.in. pola wyboru modelu, trybu predykcji, parametrów specyficznych dla inferencji oraz możliwość załadowania wag modelu z pliku.

Aby wczytać wagi modelu, użytkownik musi kliknąć przycisk **Browse** i wybrać odpowiedni plik z dysku. Wybrane wagi są następnie wykorzysty-

wane do wykonania predykcji na wczytanych obrazach.

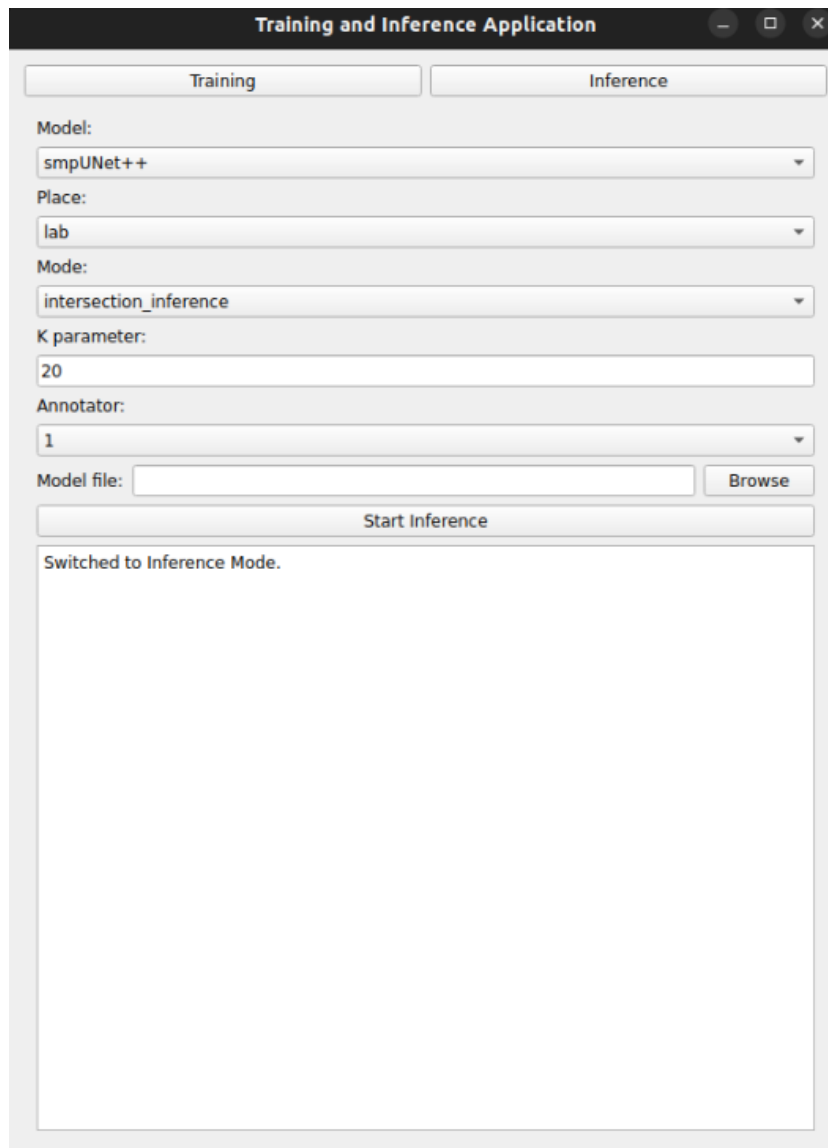


Figure 2: Widok pełnego GUI dla trybu inferencji.

2.3 Listy rozwijane

W obu trybach aplikacji `gui.py` dostępne są różne listy rozwijane, które umożliwiają użytkownikowi wybór odpowiednich opcji:

smpUNet
smpUNet++
MANet
DeepLabV3+
DualUNetPlusPlus
QuadUNetPlusPlus

Figure 3: Lista rozwijana z wyborem modelu.

CrossEntropyLoss
CrossEntropyLossWeight
BCEWithLogitsLoss
BCE

Figure 4: Lista rozwijana z wyborem funkcji straty.

Adam
SGD
RMSprop

Figure 5: Lista rozwijana z wyborem optymalizatora.

CosineAnnealingLR
ReduceLROnPlateau
MultiStepLR
None

Figure 6: Lista rozwijana z wyborem schedulera.

laptop
lab
komputer

Figure 7: Lista rozwijana z wyborem miejsca przetwarzania.

BasicAugmentation
ClassSpecificAugmentation

Figure 8: Lista rozwijana z wyborem rodzaju augmentacji.

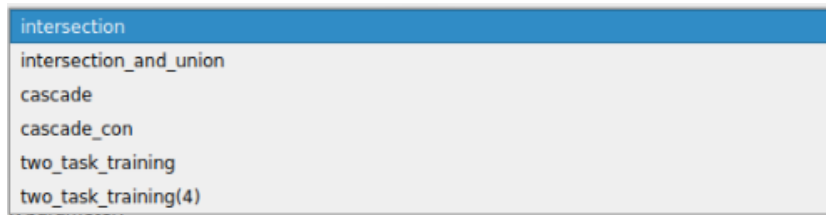


Figure 9: Lista rozwijana z wyborem rodzaju eksperymentu

2.4 Rozpoczęcie trenowania lub inferencji

Po wypełnieniu formularzy zarówno w trybie trenowania, jak i inferencji, użytkownik może nacisnąć odpowiedni przycisk (**Start Training** lub **Start Inference**), aby rozpocząć proces. Wyniki i logi są wyświetlane w dedykowanym oknie poniżej przycisków.

Aplikacja `gui.py` oferuje użytkownikowi pełną kontrolę nad procesem trenowania oraz inferencji, umożliwiając łatwe dostosowanie parametrów oraz wygodne monitorowanie wyników.

3 Aplikacja 2: gui_predict.py

Aplikacja `gui_predict.py` służy do wykonywania predykcji na obrazach przy użyciu wczytanych modeli. Umożliwia konfigurację progów predykcji, a wyniki są wyświetlane bezpośrednio w interfejsie graficznym. Aplikacja oferuje również wizualizację wyników predykcji w postaci masek na obrazach.

3.1 Wczytanie modelu i obrazu

Aby rozpocząć pracę z aplikacją, należy najpierw wybrać odpowiedni model z listy rozwijanej, a następnie wczytać zapisane wagi modelu z lokalnego dysku za pomocą przycisku **Select Model File**. Kolejnym krokiem jest wczytanie obrazu, na którym zostanie wykonana predykcja. Kliknięcie przycisku **Select Image** umożliwia załadowanie obrazu z lokalnego dysku.

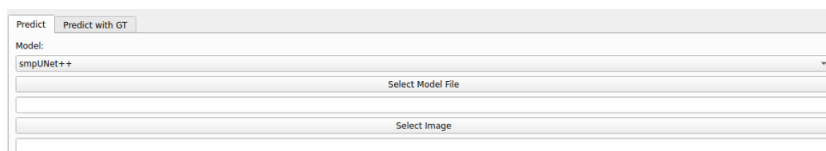


Figure 10: Wczytanie modelu i obrazu w aplikacji.

3.2 Konfiguracja progów predykcji

Po wczytaniu obrazu aplikacja umożliwia konfigurację progów predykcji dla dwóch klas: **head** (głowa) oraz **tail** (ogon). Dla każdej z klas dostępny jest suwak, który pozwala na ustawienie wartości progowej w zakresie od 0 do 255. Przesunięcie suwaka powoduje automatyczne przeliczenie predykcji i aktualizację wyświetlanego wyniku w interfejsie.



Figure 11: Suwaki do konfiguracji progów predykcji.

3.3 Wykonanie predykcji

Po skonfigurowaniu progów predykcji użytkownik może uruchomić proces predykcji, klikając przycisk **Run Prediction**. Wyniki predykcji są wyświetlane w postaci kolorowej maski. Kolory masek odpowiadają poszczególnym klasom: zielony dla klasy **head**, czerwony dla klasy **tail**.



Figure 12: Wynik predykcji.

3.4 Tryb z ground truth (GT)

Aplikacja oferuje także tryb predykcji z użyciem danych referencyjnych (ground truth). W tym trybie, oprócz wczytania obrazu do analizy, użytkownik wczytuje również plik z ground truth, który zostaje wyświetlony obok wyniku predykcji. Umożliwia to porównanie wyników modelu z rzeczywistymi danymi.

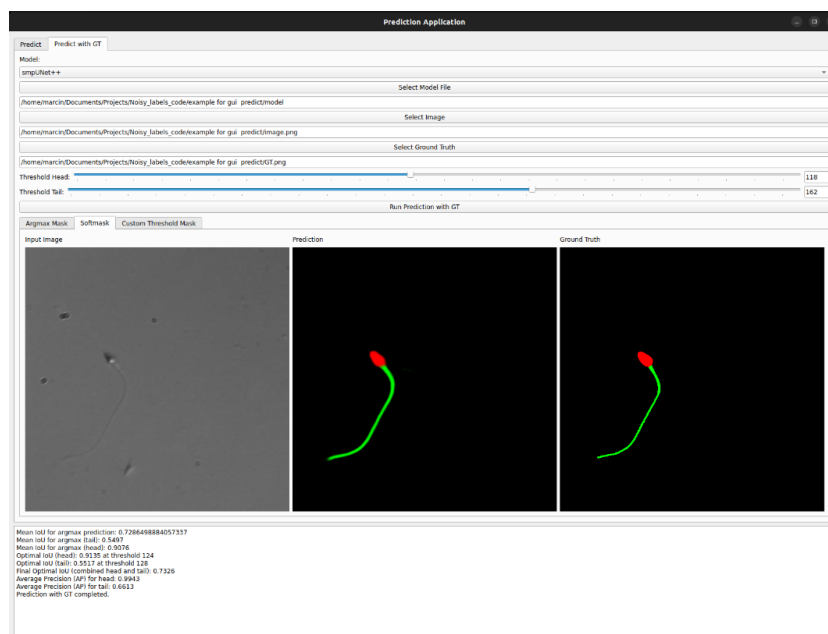


Figure 13: Tryb Predict with GT z wczytanym ground truth i wynikiem predykcji.

3.5 Zakładki z wynikami

Wyniki predykcji można przeglądać w trzech różnych zakładkach:

- **Argmax Mask** – wynik predykcji z użyciem funkcji argmax, który przypisuje każdemu pikselowi jedną klasę.
- **Softmask** – wyniki predykcji wyświetlane w postaci miękkich masek, które pokazują stopień pewności modelu co do klasyfikacji poszczególnych pikseli.
- **Custom Threshold Mask** – wynik predykcji z zastosowaniem progów konfigurowalnych przez użytkownika.

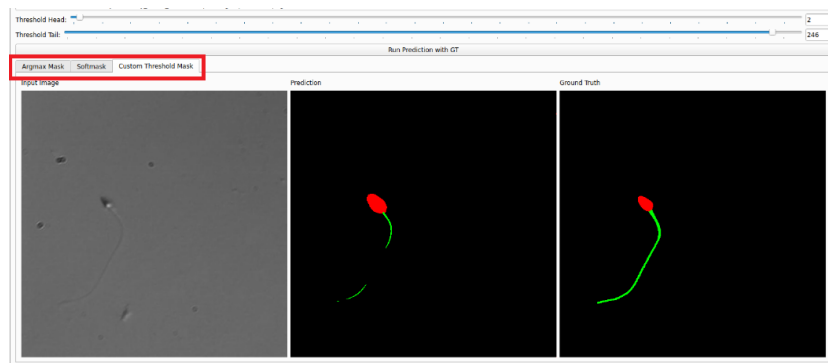


Figure 14: Zakładki z wynikami predykcji w aplikacji.

3.6 Wyświetlanie logów działania

Na dole interfejsu aplikacja wyświetla logi działania, które informują o postępach predykcji, wartościach metryk oraz ewentualnych komunikatach błędów. Logi aktualizują się na bieżąco i są pomocne w diagnozowaniu problemów podczas działania aplikacji.



Figure 15: Wyświetlanie logów działania w aplikacji.