Małe zbiory CT urazów: wpływ splitu danych na detekcję 2D i segmentację 3D oraz sanity-checki pipeline'u

Konrad Gorzelanczyk[1] Szymon Krawczyk[1] Stanisław Krzywiak[1]
[1]Politechnika Poznańska

12 lutego 2026

Streszczenie

Tło. Modele detekcji i segmentacji urazów w tomografii komputerowej (CT) są szczególnie podatne na artefakty walidacji przy małej liczbie pacjentów. Korelacja sąsiednich przekrojów w osi Z może powodować przeciek informacji przy podziale per-slice, co sztucznie zawyża metryki.

Cel. Porównać detekcję 2D i segmentację 3D przy walidacji na poziomie pacjenta oraz zaproponować minimalny zestaw sanity-checków pozwalający odróżnić błąd pipeline'u od rzeczywistego braku uogólniania.

Metody. Dla detekcji 2D porównaliśmy scenariusz z przeciekiem (split per-slice) z poprawnym scenariuszem (split per pacjent), raportując mAP oraz krzywe Precision–Recall w funkcji progu *confidence*. Dla segmentacji 3D zastosowaliśmy nnU-Net (`3d_fullres`) w schemacie LOSO (23 foldy) i raportowaliśmy Dice oraz miary błędów FP i FN. Poprawność pipeline'u zweryfikowaliśmy testem single-case overfit, kontrolą deterministyczności inferencji oraz audytem masek i próbkowania foreground.

Wyniki. Przy podziale per-slice detekcja osiągała pozornie wysoką skuteczność (F1 ≈ 0.923). Po przejściu na split per pacjent nastąpiło załamanie jakości (Precision ≈ 0.117, Recall ≈ 0.055, F1 ≈ 0.075), co wskazuje na silnie ograniczone uogólnianie między pacjentami. W segmentacji 3D uzyskano medianę Dice $= 0.0$, a 22/23 przypadki miały Dice < 0.05 (średnia 0.0054). Jednocześnie single-case overfit osiągał Dice ≈ 0.97, co potwierdza poprawność etykiet i pipeline'u oraz sugeruje, że dominującym problemem jest uogólnianie przy małym N i skrajnej nierównowadze foreground.

Wnioski. Dla małych zbiorów CT konieczna jest walidacja patient-level oraz raportowanie sanity-checków, które minimalizują ryzyko błędnej interpretacji "dobrych" wyników uzyskanych przez przeciek danych.

1 Wprowadzenie

Zadania detekcji i segmentacji urazów w CT są trudne ze względu na małe i rzadkie zmiany, dużą zmienność między pacjentami oraz silną korelację przekrojów w osi z. W przypadku uczenia na przekrojach 2D, pozornie dobre wyniki mogą być wynikiem błędnego podziału danych (np. split per-slice), natomiast w segmentacji 3D problemy mogą wynikać zarówno z generalizacji, jak i z błędów pipeline'u (np. konwersji masek, niezgodności geometrii).

Ryzyko *patient leakage* przy pracy na przekrojach 2D jest dobrze opisane w literaturze. Yagis i wsp. pokazali, że split na poziomie slice może sztucznie zawyżać wyniki, a w skrajnym teście daje bardzo wysoką skuteczność nawet dla losowo przypisanych etykiet; efekt ten znika po przejściu na podział subject-level.[2] Dlatego w tej pracy traktujemy podział na poziomie pacjenta jako warunek konieczny dla wiarygodnej oceny modeli.

Wkładem pracy jest (i) audyt walidacji na poziomie pacjenta dla detekcji 2D i segmentacji 3D, (ii) analiza pułapek metryk w detekcji, oraz (iii) protokół sanity-check, który ujawnia błąd pipeline'u, gdy model nie potrafi przeuczyć nawet danych treningowych w trybie inferencji.

2 Dane i adnotacje

2.1 Zbiór danych

Zbiór obejmuje $N = 23$ pacjentów z badaniami CT oraz adnotacjami urazów.

W wariancie 2D analizujemy łącznie 7 360 przekrojów, z czego 5 861 jest dodatnich, a 1 499 ujemnych. Slice dodatni definiujemy jako przekrój zawierający co najmniej 1 obiekt

GT (bbox) — tj. przekrój, na którym uraz jest widoczny i został oznaczony w adnotacjach 2D. Łącznie oznaczono 78 unikalnych instancji urazów (średnio 3.39 na pacjenta, zakres 2–6), przy czym każda instancja może być widoczna na wielu kolejnych slice (stąd różne rzędy wielkości: instancje 3D vs liczba adnotacji 2D per-slice). Występuje 12 klas urazów: CC, CK, CO, G, MK, MO, MS, SAH, SDH, Z, ZK, ZX; definicje skrótów oraz mapowanie na kategorie anatomiczne i identyfikatory etykiet w modelu wieloklasowym zestawiono w Tabeli 1.

W tej pracy prowadzimy **dwa równoległe strumienie eksperymentów** oparte o tę samą kohortę 23 pacjentów, ale o **różnej jednostce obserwacji i różnych etykietach**: (i) detekcję 2D/2.5D na poziomie pojedynczego przekroju (slice) z ramkami bbox oraz (ii) segmentację 3D na poziomie wolumenu z maską binarną (uraz vs tło). Strumienie te były prowadzone równolegle po to, aby sprawdzić, czy przy małym N i rzadkim foreground któryrkolwiek wariant (2D lub 3D) daje stabilniejsze wyniki oraz aby zdiagnozować, czy obserwowane porażki wynikają z błędów walidacji/pipeline'u, czy z realnego braku generalizacji.

W związku z tym **nie mieszamy "światów" 2D i 3D**: wszędzie jawnie podajemy (a) jednostkę obserwacji (slice vs pacjent/wolumen), (b) protokół walidacji (per-slice vs patient-level/LOOCV/LOSO) oraz (c) odpowiadające im statystyki danych i metryki.

Dataset card (wersja 1): detekcja 2D (YOLO; bbox per slice)

Cel zadania. Detekcja (binarna lub wieloklasowa) urazów na pojedynczych przekrojach CT.

Jednostka obserwacji (*sample*). Pojedynczy przekrój CT (2D) po preprocessingu (windowing, resize/letterbox itp.).

Co jest etykietą i co jest "obiektem GT".

- **Etykieta klasy (multi-class):** jedna z 12 klas {CC, CK, CO, G, MK, MO, MS, SAH, SDH, Z, ZK, ZX}.
- **Obiekt GT (w ewaluacji detekcji):** pojedynczy bounding box na danym slice (tj. jedna ramka bbox = (x_1, y_1, x_2, y_2)) z przypisaną klasą. W detekcji "liczba GT obiektów" oznacza zatem **liczbę ramek bbox** w zbiorze ewaluacyjnym (a nie liczbę pacjentów ani liczbę instancji 3D).

Definicja "pozytywu".

- **Pozytywny sample (positive slice):** slice, na którym występuje co najmniej 1 obiekt GT (co najmniej 1 bbox) *lub* (w definicji spójnej z 3D) slice z co najmniej 1 voxel foreground w masce 3D.
- **Pozytywny obiekt (positive object):** pojedynczy obiekt GT (bbox) na danym slice.

Kluczowe liczby (nasza kohorta).

- $N = 23$ pacjentów.
- 7 360 slice łącznie; 5 861 slice dodatnich (tzn. z co najmniej 1 bbox w adnotacji 2D).
- **78 instancji urazów** *w sensie kliniczno-3D* (instancje/zmiany liczone per pacjent w wolumenie; patrz karta 3D).
- Dodatkowo utrzymujemy wariant "pojedynczy reprezentatywny przekrój na uraz" (plik `locations.csv`; 2 399 rekordów), użyteczny do analiz pomocniczych; główny trening/ewaluacja YOLO korzysta z listy wszystkich przekrojów z widocznymi urazami (`urazy_full_list.csv`; 5 861 rekordów).

Uwaga: konflikt "78 vs 2475" (i gdzie wchodzą TP/FP/FN).

- **78** odnosi się do liczby *unikalnych instancji urazów* w wolumenach ("ile zmian w kohorcie").
- **2475** w Tabeli 5 to **liczba TP** (trafionych bbox) w scenariuszu *z przeciekiem* (split per-slice) — to **nie** jest liczba instancji 3D.
- W tym samym scenariuszu liczba obiektów GT (bbox) w ewaluacji wynosi $\mathrm{GT} = \mathrm{TP} + \mathrm{FN} = 2475 + 91 = 2566$ (Tabela 5).
- Różnica skali bierze się stąd, że jedna instancja urazu 3D może obejmować wiele slice i w danych 2D jest reprezentowana przez wiele ramek bbox (po jednej na slice).

Dataset card (wersja 2): segmentacja 3D (nnU-Net; maska binarna per wolumen)

Cel zadania. Segmentacja binarna uraz vs tło w 3D (wolumen CT).

Jednostka obserwacji (*sample*). Pojedynczy wolumen CT jednego pacjenta (po preprocessingu nnU-Net: resampling + normalizacja).

Co jest etykietą i co jest "GT".

- **GT w segmentacji:** binarna maska 3D $\in \{0, 1\}$ w tej samej geometrii co wolumen (injury vs background).
- **"Liczba GT obiektów" w 3D:** w tym strumieniu zadanie jest *maskowe* (voxel-wise), więc podstawową

Skrót	Nazwa kliniczna (EN)	Nazwa kliniczna (PL)	Kategoria	ID
MS	Cerebral contusion	Stłuczenie mózgu	Brain	1
MK	Brain haematoma	Krwiak mózgu	Brain	2
SAH	Subarachnoid haemorrhage	Krwotok podpajęczynówkowy	Meningeal hemorrhage	3
SDH	Subdural haemorrhage	Krwiak podtwardówkowy	Meningeal hemorrhage	4
CC	Scalp laceration	Rozerwanie skóry głowy	Scalp	5
CK	Scalp contusion	Stłuczenie skóry głowy	Scalp	6
ZK	Fracture with haematoma	Złamanie z krwiakiem	Skull fracture	7
Z	Fracture	Złamanie kości czaszki	Skull fracture	8
ZX	Displaced fracture	Złamanie z przemieszczeniem	Skull fracture	9
CO	Subcutaneous emphysema	Odma podskórna	Air	10
MO	Intracranial emphysema (pneumocephalus)	Odma wewnątrzczaszkowa	Air	11
G	Glial scar (gliotic scar)	Blizna glejowa	Scar	12

Tabela 1: Definicje klas i skrótów urazów oraz mapowanie na kategorie anatomiczne i identyfikatory etykiet w modelu wieloklasowym (ID=0 oznacza tło).

Miara	Wartość
Liczba pacjentów (N)	23
Liczba badań CT	23
łączna liczba slice	7 360
Slice dodatnie / ujemne	5 961 / 1 399
Liczba instancji urazów (łącznie)	78
Spacing $x/y/z$ (zakresy) [mm]	0.432–0.576 / 0.432–0.576 / 0.625–0.665

Tabela 2: Statystyki zbioru danych: "twarde minimum" do głównego tekstu. Szczegóły (częstości per-klasa, rozkłady objętości urazów i miary rzadkości foreground) podano w Aneksie C.

jednostką prawdy jest **zbíor voxel-i foreground**, a nie bbox. Jeśli potrzebna jest liczba obiektów, można ją zdefiniować jako liczbę instancji (np. składowych spójnych) w masce; w naszej kohorcie odpowiada temu liczba **78 instancji urazów** raportowana powyżej.

Definicja "pozytywu".

- **Pozytywny sample (positive case):** pacjent z niepustą maską (co najmniej 1 voxel foreground). W tej kohorcie wszystkie 23/23 przypadki mają niepusty foreground (audyt wartości masek).

Kluczowe liczby (nasza kohorta).

- $N = 23$ wolumeny (po 1 na pacjenta).
- Maski binarne {0,1}; foreground ekstremalnie rzadki (średnio 0.0126% voxel-i).

W głównym tekście raportujemy wyłącznie "twarde minimum" (Tabela 2), natomiast rozkłady per pacjent/per slice, częstości per-klasa oraz statystyki objętości urazów przeniesiono do Aneksu C.

Kontekst kliniczny kohorty. Zgodnie z informacją przekazaną przez radiologa podczas pracy nad adnotacjami (radiolog, komunikacja osobista, 2025), badania mogły pochodzić od pacjentów wielokrotnie obrazowanych w przebiegu urazu, a część zmian mogła mieć charakter subtelny i trudny do oceny w standardowym odczycie. W praktyce może to oznaczać obciążenie zbioru w kierunku przypadków "trudnych" oraz ograniczoną czytelność urazów w pojedynczym badaniu.

Rozkład klas urazów raportujemy na poziomie pacjenta (nie per slice), jako liczbę pacjentów z daną klasą oraz liczbę dodatnich sliceów na pacjenta, ponieważ sąsiednie slice w osi Z są silnie skorelowane.

2.2 Format danych i adnotacje

W detekcji wykorzystujemy pliki CSV o schemacie: `case, x1, y1, x2, y2,label,mirror,z`, gdzie `mirror` oznacza augmentację, a `z` jest indeksem przekroju.

Procedura adnotacji i kontrola jakości etykiet. Adnotacje przygotował radiolog. Oznaczenia nie były weryfikowane przez drugiego eksperta, a sporne przypadki rozwiązywano ad hoc (bez formalnego protokołu arbitrażu). W detekcji 2D dysponujemy dwiema reprezentacjami adnotacji: (i) `locations.csv` zawiera jeden reprezentatywny przekrój (pojedynczy bbox) dla każdego wpisu urazu (2 399 rekordów), natomiast (ii) `urazy_full_list.csv` zawiera

listę wszystkich przekrojów, na których dany uraz jest widoczny i został oznaczony (5 861 rekordów). W głównych eksperymentach YOLO (trening/ewaluacja per-slice) wykorzystujemy `urazy_full_list.csv`, tj. bbox są zdefiniowane na każdym slice z widocznym urazem, a nie wyłącznie na pojedynczym przekroju.

Wszystkie transformacje geometryczne obrazu, w szczególności resize oraz ewentualny letterbox/padding, są deterministycznie odtwarzane dla bbox. Dla każdej próbki logujemy parametry transformacji (skala, przesunięcie) i weryfikujemy poprawność przez losową inspekcję wizualną (obraz z nałożonym bbox po wszystkich transformacjach). Ten krok jest konieczny, ponieważ nawet niewielki błąd w mapowaniu współrzędnych prowadzi do sztucznego wzrostu FN i pozornego "braku generalizacji".

2.3 Preprocessing CT

Preprocessing obejmuje konwersję DICOM do skali Hounsfielda (HU), a następnie standardowy windowing (C/W). W wariancie 2.5D (multi-window) stosujemy trzy **stałe** presety okien jako trzy kanały wejściowe (np. brain, subdural, bone). Każdy kanał jest obcinany do $[C - W/2,\ C + W/2]$ i normalizowany do zakresu $[0, 1]$.

Dla segmentacji 3D wykorzystano standardowy preprocessing nnU-Net (fingerprinting danych, resampling i normalizacja intensywności).

2.4 Ryzyka jakości danych

W pracy identyfikujemy ryzyko błędów geometrii i konwersji masek (np. przesunięcia przy TIF→NIfTI), co może prowadzić do zaniżania wyników i fałszywej interpretacji jako "słaba generalizacja".

Dodatkowo uwzględniamy potencjalne ryzyko kliniczno-selekcyjne: kohorta może być wzbogacona o przypadki trudne (w tym zmiany subtelne/"occult"), co utrudnia uczenie i interpretację wyników przy małym N.

3 Metody

3.1 Protokół walidacji i splity

Stosujemy podział danych na poziomie pacjenta, aby uniknąć *patient leakage*. Dodatkowo raportujemy wyniki w protokołach LOOCV/LOSO w celu maksymalizacji wykorzystania małego zbioru.

Projekt walidacji opieramy na rekomendacjach dotyczących CV w obrazowaniu medycznym, ze szczególnym naciskiem na rozdział danych na poziomie pacjenta oraz jednoznaczne raportowanie jednostki niezależnej obserwacji.[3] Dodatkowo unikamy sytuacji "użycia danych dwa razy" (strojenie i ocena na tych samych foldach bez separacji zewnętrznej), która może prowadzić do nadmiernie optymistycznych estymacji błędu.[4]

3.2 Dane i etykiety (minimum raportowe)

Minimum raportowe: (i) łącznie 7 360 slice, (ii) 5 861 slice dodatnich i 1 499 ujemnych (definicja 2D: obecność co najmniej 1 bbox na danym slice), (iii) 78 unikalnych instancji urazów w całej kohorcie, (iv) 12 klas urazów (CC, CK, CO, G, MK, MO, MS, SAH, SDH, Z, ZK, ZX) z częstościami raportowanymi per pacjent, (v) wieloetykietowość: pacjenci mają 2–6 klas, średnio 3.39 klasy na pacjenta. Dla 3D rozkład foreground jest skrajnie rzadki na poziomie wokseli: frakcja voxeli foreground w wolumenie wynosi średnio 0.0126%.

Etykiety bbox mają postać `(x1, y1, x2, y2)` w układzie współrzędnych obrazu po preprocessing (np. resize/letterbox); opisujemy regułę mapowania współrzędnych podczas skalowania.

3.3 Preprocessing DICOM i windowing

Preprocessing obejmuje konwersję z DICOM do HU oraz standardowy windowing (C/W). W przypadku podejścia 2.5D (multi-window) stosujemy trzy **stałe** presety windowingu jako trzy kanały wejściowe (brain, subdural, bone); szczegółowe wartości podano w Aneksie (Sekcja D). Każdy kanał jest obcinany do $[C - W/2,\ C + W/2]$ i normalizowany do zakresu $[0, 1]$.

3.4 Strategia splitu i kontrola leaku

Stosujemy dwa scenariusze walidacyjne:

1) **Baseline błędny (dla demonstracji artefaktów):** split per-slice (bez grupowania po pacjencie), co może prowadzić do przecieku pacjentowego.
2) **Baseline poprawny:** split per pacjent (docelowo: `MultilabelStratified ShuffleSplit` dla stratyfikacji wieloetykietowej), oraz protokoły LOOCV/LOSO, gdzie jednostką jest pacjent.

Dla stratyfikacji wieloetykietowej (jeśli używana) próbką jest pacjent, a etykietą — wektor obecności klas na poziomie pacjenta.

Brak przecieku weryfikujemy proceduralnie: (i) listy identyfikatorów pacjentów w train/val są rozłączne, (ii) brak duplikatów plików pomiędzy splitami, (iii) logujemy konfigurację splitu jako artefakt eksperymentu.

3.5 Ewaluacja (metryki) i poprawna interpretacja

Detekcja 2D. Raportujemy mAP@0.5 i mAP@0.5:0.95 oraz Precision/Recall/F1 (dla ustalonego progu confidence i kryterium trafienia opartego o IoU). **W detekcji nie raportujemy TN; accuracy pomijamy**. Punkt pracy modelu definiujemy przez limit FP/patient, raportując czułość dla kilku poziomów FP/patient (FROC).

3.5.1 Parametry ewaluacji detekcji 2D (IoU, matching, NMS)

Parametr	Wartość	Opis
IoU threshold dla TP (mAP)	0.50:0.95	10 wartości od 0.50 do 0.95 (krok 0.05)
IoU threshold dla mAP50	0.50	Pojedynczy próg
Matching predykcji do GT	Greedy	Sortowanie po *confidence*; 1 predykcja może być dopasowana do maks. 1 obiektu GT (i odwrotnie)
NMS IoU threshold	0.45	Domyślnie 0.7, ale nadpisane w kodzie (`iou=0.45`)
Confidence threshold (train/val)	0.001	Niski próg dla pełnej krzywej Precision–Recall (`conf=0.001`)
`max_det`	3	Maksymalnie 3 detekcje na obraz (`max_det=3`)
`agnostic_nms`	False	NMS per-class (`agnostic_nms=False`)

Tabela 3: Parametry ewaluacji detekcji 2D: progi IoU, zasady dopasowania predykcji do obiektów GT oraz ustawienia NMS i progów confidence.

Segmentacja 3D. Raportujemy Dice oraz voxel-wise precision/recall, a dodatkowo objętości błędów FP/FN w mm^3.

3.6 Modele

3.6.1 Detekcja 2D

W detekcji 2D wykorzystujemy modele YOLO jako główną architekturę detekcyjną (warianty: `yolov8m`, `yolo11m`).

Dla YOLO cytujemy wersję narzędzia i repozytorium (np. Ultralytics), ponieważ nie zawsze istnieje odpowiadająca praca naukowa dla konkretnej wersji implementacji.[1]

Augmentacje oraz parametry treningu dla YOLO raportujemy w Sekcji 7 (Reproducibility), wraz z wersją biblioteki i identyfikatorem konfiguracji eksperymentu.

Podejście 2.5D w tym kontekście oznacza wariant *multi-window*: 3 kanały wejściowe odpowiadają 3 stałym oknom (C, W) zastosowanym do tego samego przekroju.

Szczegółowe parametry treningu i inferencji (w tym zakres progów *confidence* oraz IoU dla NMS) podano w Sekcji 7.

Tryby etykietowania (binary vs multi-class). W detekcji 2D raportujemy dwa rozłączne ustawienia eksperymentalne: (i) **binary** (wszystkie klasy urazów scalone; `single_cls=True`) oraz (ii) **multi-class** (12 klas trenowanych osobno; `single_cls=False`). Odpowiadające im ustawienia zostały jawnie rozdzielone w Aneksie (Tabele 14 i 15), a w Sekcji Wyniki każdy rysunek/macierze pomyłek oznaczamy, którego trybu dotyczą.

3.6.2 Segmentacja 3D

Segmentację 3D realizujemy w nnU-Net (konfiguracja `3d_fullres`, architektura `ResidualEncoderUNet`; planner: `nnUNetPlannerResEncM`) dla zadania binarnego uraz vs tło.

nnU-Net został wybrany jako punkt odniesienia, ponieważ jest szeroko stosowanym, samo-konfigurującym podejściem do segmentacji biomedycznej, które automatyzuje kluczowe decyzje treningowe i preprocessingowe.[5] Dzięki temu ograniczamy liczbę ręcznych wyborów, co jest istotne w reżimie małego N.

3.6.3 Segmentacja 3D: konfiguracja i protokół treningu (nnU-Net v2)

Framework i walidacja. Framework: nnU-Net v2. Protokół walidacji: LOSO (Leave-One-Subject-Out), 23 foldy, gdzie w każdym foldzie 1 pacjent jest walidacją/testem, a pozostałe 22 pacjenty stanowią zbiór treningowy.

Preprocessing (nnU-Net). Użyto standardowego preprocessingu nnU-Net v2 (fingerprinting danych, resampling do medianowego spacing oraz normalizacja intensywności). Parametry podano w Tabeli 4.[1]

Test-Time Augmentation (inferencja). W inferencji stosowano TTA: mirroring axes (0, 1, 2), tile step size = 0.5 oraz Gaussian smoothing: włączone.

Konfiguracja raportowana w wynikach finalnych. Parametry treningu i optymalizacji oraz ustawienia sieci/próbkowania zestawiono w Tabeli 4.

Parametr	Wartość
Konfiguracja	`3d_fullres`
Architektura	ResidualEncoderUNet
Planner	`nnUNetPlannerResEncM`
Liczba epok	1000
Optimizer	SGD, lr 0.01, momentum 0.99, Nesterov: on, weight decay 3×10^{-5}
Scheduler	PolyLR
Iterations per epoch	250
Validation iterations	50
Patch size	[96, 160, 160]
Batch size	2
Deep supervision	włączone
Foreground oversampling	0.33
Loss	Dice + Cross-Entropy (`DC_and_CE_loss`); Dice: `Memory-EfficientSoftDiceLoss`; Batch Dice: True

Tabela 4: Konfiguracja i protokół treningu dla segmentacji 3D (nnU-Net v2) raportowane w wynikach finalnych.

Augmentacje. W treningu nnU-Net stosowano domyślne augmentacje 3D: rotacje, skalowanie, elastyczne deformacje, mirroring względem osi, modyfikacje jasności i gamma, szum i rozmycie Gaussa oraz symulację niskiej rozdzielczości. Szczegółowe parametry augmentacji zestawiono w Aneksie (Sekcja E).

Ze względu na nierównowagę foreground rozważaliśmy również warianty funkcji straty oparte o indeks Tversky (w tym focal Tversky), jednak w naszych próbach nie przyniosły powtarzalnej poprawy, dlatego nie raportujemy ich wyników.[12]

[1] Szczegóły techniczne (np. dokładne ustawienia interpolacji i przetwarzania masek w `resize_segmentation`/one-hot) pozostawiono poza głównym tekstem jako element reprodukowalności.

3.7 Metryki i ich pułapki

W zadaniach z bardzo małymi, niepewnymi lub rzadkimi zmianami metryki nakładania (np. Dice) mogą być niestabilne i silnie zależne od objętości referencji, a także nie karzą wprost fałszywych alarmów w "pustych" przypadkach.[6] Dlatego oprócz Dice raportujemy metryki uzupełniające zgodne z przeglądem klasycznych miar segmentacji 3D (m.in. dystanse powierzchniowe).[7] Dla komponentu detekcyjnego (FP/scan przy zadanym poziomie czułości) sensownym standardem jest analiza FROC, powszechnie stosowana w detekcji zmian w obrazowaniu medycznym.[8]

3.7.1 Detekcja

Raportujemy mAP@0.5 oraz mAP@0.5:0.95, a także Precision, Recall i F1 w funkcji progu confidence.

3.7.2 Segmentacja

Dla segmentacji raportujemy Dice oraz voxel-wise precision/recall, a także wolumeny błędów FP/FN w mm^3. Obliczono voxel-wise macierz pomyłek dla segmentacji binarnej (injury vs background). Dla każdego folda LOOCV zliczano liczbę wokseli TP, FP, FN oraz TN w przestrzeni predykcji zgodnej z geometrią GT, a następnie sumowano te zliczenia po wszystkich 23 foldach. Oprócz macierzy w postaci surowych zliczeń przedstawiono wersję znormalizowaną względem klasy prawdziwej (kolumny), co pozwala bezpośrednio interpretować czułość (TPR) i swoistość (TNR) przy ekstremalnej nierównowadze klas. Do wizualizacji zastosowano skalę logarytmiczną, aby komórki o małych wartościach były czytelne.

3.8 Protokół audytu (sanity-check)

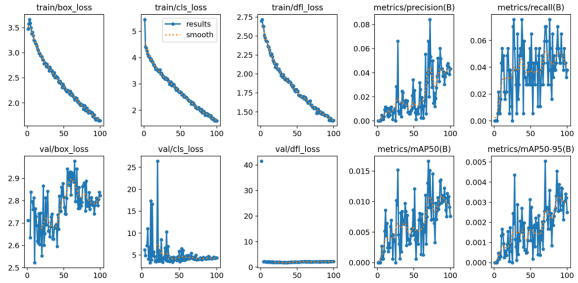
Zastosowano predefiniowany zestaw sanity-check (Aneks A, Tabela 8). Celem było wykluczenie typowych awarii pipeline'u.

4 Wyniki

Ten fragment dotyczy wyłącznie detekcji 2D/2.5D (YOLO). Segmentacja 3D (nnU-Net) stanowi osobny strumień prac opisany w oddzielnej sekcji.

4.1 Leakage experiment: binary setting (`single_cls=True`)

W celu pokazania, jak łatwo uzyskać pozornie bardzo dobre metryki w danych CT silnie skorelowanych po osi z, raportujemy wyniki dla błędnego splitu per-slice (bez grupowania po pacjencie), który dopuszcza przeciek pacjentowy. Podkreślamy, że "bardzo dobre wyniki" w tym ustawieniu są artefaktem metodologicznym.

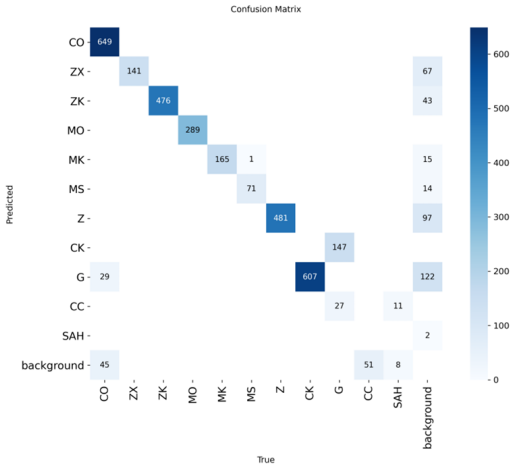


Rysunek 1: Wyniki YOLO w konfiguracji splitu, która dopuszcza przeciek pacjentowy (split per-slice): krzywe treningu oraz binarna macierz pomyłek (injury vs background). Ustawienie **binary**: `single_cls=True` (wszystkie klasy scalone). Ten wynik traktujemy jako artefakt metodologiczny i punkt odniesienia do demonstracji problemu leaku.

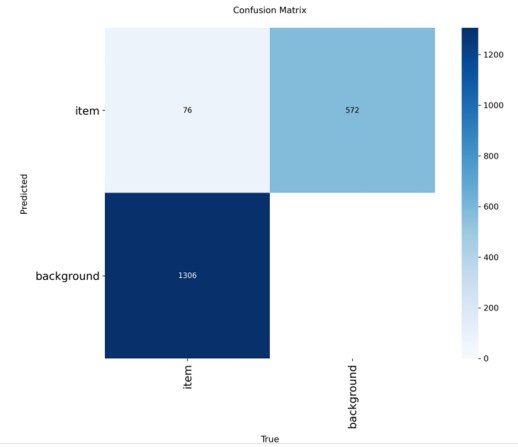
4.2 Leakage experiment: multi-class setting (`single_cls=False`)

W macierzy wieloklasowej elementy na przekątnej odpowiadają poprawnym detekcjom danej klasy, natomiast elementy poza przekątną pokazują dominujące pomyłki pomiędzy klasami urazów. Poza błędami związanymi z tłem, największe pomyłki między klasami dotyczą przypadków, w których klasa CK bywa przewidywana jako G (607 przypadków), a także gdy klasa G jest przewidywana jako CK (147 przypadków).

Ostatni wiersz `predicted background` należy interpretować jako FN (przypadki, w których model nie wykrył obiektu), a ostatnią kolumnę `true background` jako FP (fałszywe alarmy). TN nie raportujemy w detekcji; accuracy pomijamy.



Rysunek 2: Wieloklasowa macierz pomyłek dla detekcji. Ustawienie **multi-class**: `single_cls=False` (12 klas). Ostatnia kolumna "true background" odpowiada FP, a ostatni wiersz "predicted background" odpowiada FN. TN nie raportujemy w detekcji; accuracy pomijamy.



Rysunek 3: Macierz pomyłek po zastosowaniu splitu per pacjent (bez leaku). Ustawienie **binary**: `single_cls=True` (wszystkie klasy scalone). Widoczna dominacja FN i FP oraz spadek skuteczności, co wskazuje na ograniczenia wynikające z małej liczby niezależnych pacjentów i dużej korelacji przekrojów.

4.3 Patient-level split (no leakage): performance collapse and data limitation

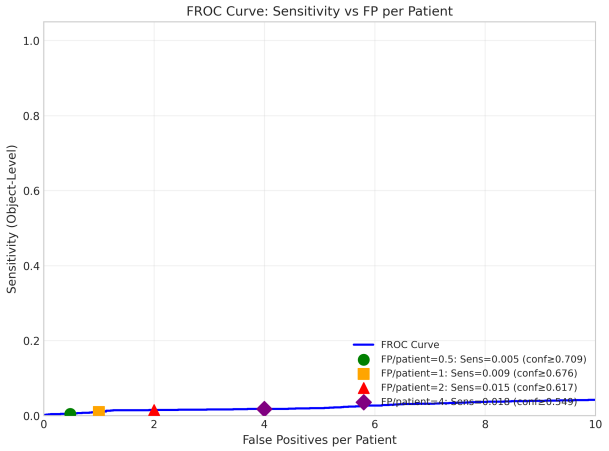
Dla splitu per pacjent (bez przecieku) obserwujemy istotny spadek skuteczności i dominację błędów FN/FP. Zliczenia TP/FP/FN dla obu scenariuszy zestawiono w Tabeli 5; interpretacja błędów związanych z tłem znajduje się w Sekcji Metody.

Split	TP	FP	FN
Per-slice (z przeciekiem)	2475	320	91
Per pacjent (bez przecieku)	76	572	1306

Tabela 5: Zliczenia TP/FP/FN dla detekcji 2D w scenariuszu z przeciekiem (split per-slice) oraz w walidacji patient-level (split per pacjent). Jednostką zliczeń jest obiekt GT (bbox) na pojedynczym obrazie 2D (slice), tj. suma po wszystkich ewaluowanych slice w danym scenariuszu.

4.3.1 YOLO LOSO: FROC i punkt pracy

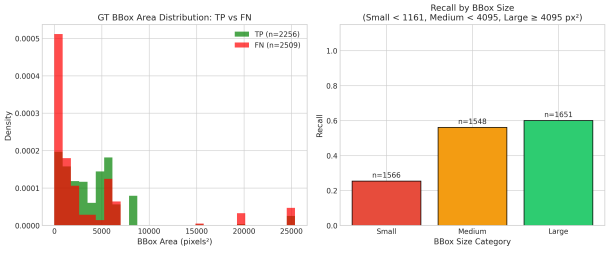
Dodatkowo oceniliśmy detekcję w protokole leave-one-subject-out (LOSO, $N = 23$), gdzie w każdym foldzie jeden pacjent stanowił walidację. Jako główną miarę raportujemy krzywą FROC, czyli czułość obiektową w funkcji liczby fałszywych detekcji na pacjenta (FP/patient), ponieważ jest to bardziej interpretowalne klinicznie niż pojedyncza wartość mAP. W zakresie FP/patient ≤ 2 uzyskana czułość obiektowa była bardzo niska, co wskazuje na brak stabilnej generalizacji w reżimie patient-level przy małej liczbie pacjentów.



Rysunek 4: FROC dla YOLO w protokole LOSO ($N = 23$). Oś X: FP/patient, oś Y: czułość obiektowa (TP/GT) liczona przez dopasowanie bbox z kryterium IoU ≥ 0.5. Zaznaczono punkty pracy odpowiadające wybranym limitom FP/patient (np. 1 i 2) oraz odpowiadające im progi *confidence*.

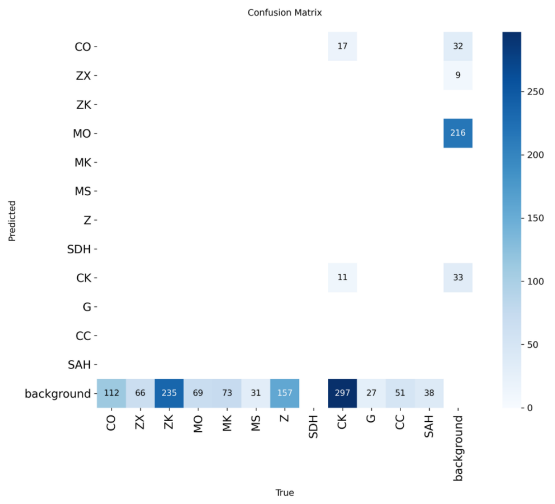
4.3.2 Failure mode: wpływ rozmiaru zmiany

Aby zidentyfikować dominujący tryb porażki, przeanalizowaliśmy czułość w zależności od rozmiaru adnotacji (pole bbox). Wyniki wskazują na istotny spadek czułości dla małych zmian, co jest spójne z silną nierównowagą klas i bardzo małym sygnałem w danych. W praktyce oznacza to, że model częściej wykrywa większe, bardziej rozlane urazy, a małe ogniska pozostają niewidoczne w reżimie patient-level.



Rysunek 5: Czułość (TP/GT) w funkcji rozmiaru bbox GT. Obiekty podzielono na kategorie rozmiaru na podstawie pola bbox. Dopasowanie bbox z kryterium IoU ≥ 0.5.

Rys. 6 przedstawia wieloklasową macierz pomyłek dla detekcji w ustawieniu patient-level po poprawie. Dominacja wiersza `predicted background` wskazuje, że głównym trybem porażki są niewykrycia obiektów (FN) we wszystkich klasach, a nie mylenie typów urazów między sobą. Ostatni wiersz `predicted background` należy interpretować jako FN, a ostatnią kolumnę `true background` jako FP.



Rysunek 6: Wieloklasowa macierz pomyłek YOLO w walidacji patient-level (bez przecieku). Ustawienie **multi-class**: `single_cls=False` (12 klas).

4.3.3 Ablacja: rozmiar modelu, augmentacje i downsampling

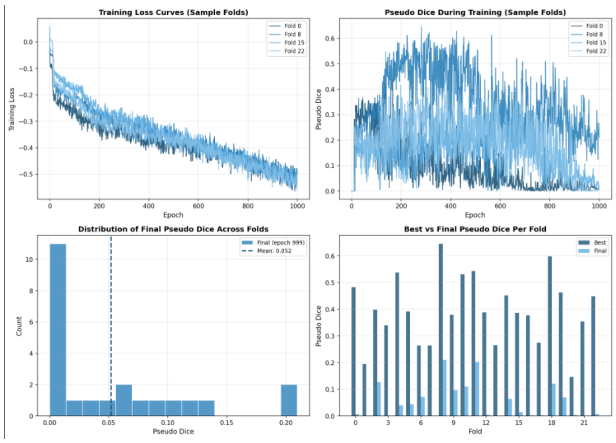
Po poprawieniu splitu na poziomie pacjenta, zmiana rozmiaru modelu YOLO, polityki augmentacji oraz zwiększenie liczby przekrojów (cofnięcie downsamplingu) nie doprowadziły do istotnej poprawy metryk, co wskazuje na dominujące ograniczenie liczby i różnorodności niezależnych pacjentów.

4.4 Segmentacja 3D: sanity-check (nnU-Net LOOCV/LOSO)

W protokole LOSO (23 foldy, jeden pacjent jako test) rozkład Dice dla klasy uraz jest skrajnie skośny z dominacją przypadków bez nakładania (zero overlap); podsumowanie statystyk przedstawiono w Tabeli 6.

Protokół	Mean	SD	Median	IQR	Min	Max
LOSO (Dice)	0.0054	0.0231	0.0000	[0.0000, 0.0000]	0.0000	0.1147

Tabela 6: Podsumowanie rozkładu Dice dla segmentacji 3D (klasa uraz) w protokole LOSO.



Rysunek 7: Statystyki treningu nnU-Net 3D (LOSO, 23 foldy). Górny lewy panel: krzywe training loss w funkcji epoki dla przykładowych foldów. Górny prawy panel: pseudo-Dice raportowany w trakcie treningu dla tych samych foldów. Dolny lewy panel: rozkład końcowego pseudo-Dice (epoka 999) dla wszystkich foldów, z zaznaczoną średnią. Dolny prawy panel: porównanie pseudo-Dice dla checkpointów `best` i `final` w każdym foldzie.

4.4.1 Stabilność treningu i checkpointy

Podsumowanie rozkładu Dice dla LOSO przedstawiono powyżej oraz w Tabeli 6.

Rys. 7 przedstawia statystyki treningu nnU-Net 3D w protokole LOSO. Krzywe funkcji straty (loss) systematycznie obniżają się w trakcie treningu w pokazanych foldach, co wskazuje na stabilną optymalizację i brak oczywistej awarii procesu uczenia jako takiego.

Jednocześnie metryka pseudo-Dice raportowana w trakcie treningu jest silnie niestabilna między foldami oraz w czasie. Rozkład końcowego pseudo-Dice (epoka 999) jest mocno skośny z dominacją wartości bliskich zera oraz nielicznymi foldami o wyższych wartościach. Dodatkowo w większości foldów checkpoint `best` osiąga wyższy pseudo-Dice niż checkpoint `final`, co sugeruje, że późna faza treningu nie poprawia jakości segmentacji i może prowadzić do pogorszenia metryki walidacyjnej. Te obserwacje wspierają wniosek, że problemem jest ograniczona generalizacja w reżimie małego N, a nie prosty błąd pipeline'u.

Pseudo-Dice traktujemy jako metrykę diagnostyczną z logów treningowych nnU-Net i nie utożsamiamy jej bezpośrednio z końcowym Dice liczonym po pełnej inferencji wolumenowej w LOOCV/LOSO.

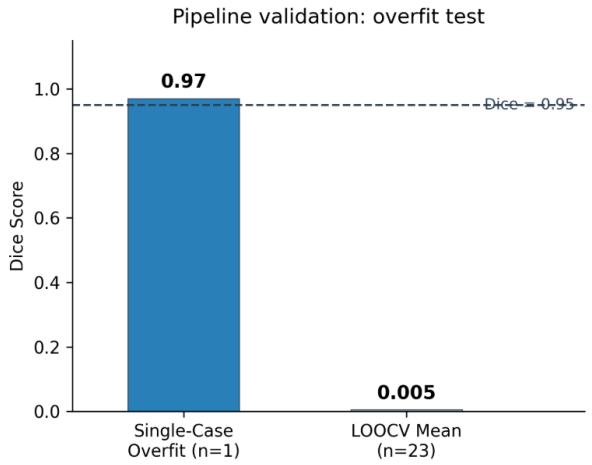
Wyniki sanity-check (zdefiniowane w Metodach; Tabela 8)

wskazują, że: (i) powtórzona inferencja z TTA off dała identyczne wyniki w 11 foldach, (ii) `class_locations` było niepuste dla 23/23 pacjentów, oraz (iii) Dice liczone w przestrzeni preprocessingu bywało równe 0 mimo niepustych predykcji.

Test przeuczenia na pojedynczym przypadku (single-case overfit) osiągnął Dice ≈ 0.97, co pokazuje, że pipeline jest zdolny do dopasowania segmentacji w kontrolowanym ustawieniu, natomiast problem ujawnia się przy walidacji patient-level LOSO, gdzie występuje silna heterogeniczność między pacjentami i mała liczebność zbioru.

4.4.2 Pipeline validation: single-case overfit

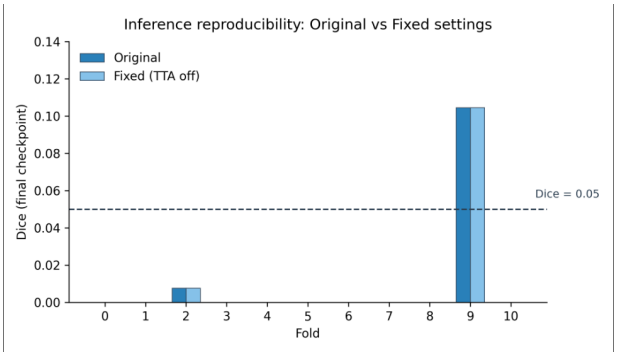
Model osiągnął Dice ≈ 0.97 w teście przeuczenia na pojedynczym przypadku ($n = 1$), podczas gdy **średnia Dice w LOOCV/LOSO** ($n = 23$) wynosi ≈ 0.005. Wynik ten wskazuje, że pipeline jest zdolny do nauczenia się segmentacji dla konkretnego przypadku, natomiast w protokole patient-level LOOCV/LOSO występuje praktyczny zanik generalizacji między pacjentami.



Rysunek 8: Pipeline validation poprzez single-case overfit. Dice w teście przeuczenia na jednym pacjencie ($n = 1$) w porównaniu do **średniej** Dice w protokole LOOCV/LOSO ($n = 23$). Wysoki Dice w overfit potwierdza poprawność pipeline'u dla danego przypadku, natomiast bardzo niski wynik LOOCV wskazuje na brak generalizacji między pacjentami.

4.4.3 Deterministyczność inferencji (lock inference)

W 11 foldach ponowna inferencja z zablokowanymi ustawieniami dała identyczne wyniki jak pierwotna ewaluacja. Zgodność wyniosła 11/11 dla checkpointu `best` oraz 11/11 dla checkpointu `final`, a **średnia różnica Dice** była równa 0.0000. Wynik ten wyklucza wpływ losowości inferencji oraz TTA na obserwowany spadek skuteczności segmentacji.



Rysunek 9: Powtarzalność inferencji nnU-Net (lock inference). Porównanie wyników Dice dla inferencji oryginalnej oraz inferencji z zablokowanymi ustawieniami (TTA off) dla 11 foldów. Wartości są identyczne w obu wariantach, co wskazuje na deterministyczność ewaluacji. Linia przerywana oznacza próg Dice = 0.05.

4.4.4 Audyt próbkowania foreground (nnU-Net `class_locations`)

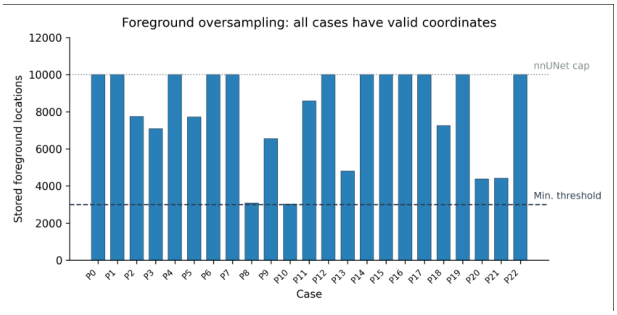
Audyt próbkowania foreground potwierdził poprawność `class_locations` dla wszystkich przypadków: 23/23 miało niepuste `class_locations` dla klasy urazu (min 3 032; średnia 8 028; max 10 000).

4.4.5 Audyt wartości etykiet (label value audit)

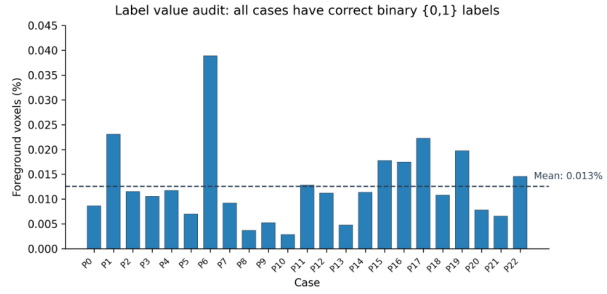
Audyt wartości etykiet potwierdził, że wszystkie przypadki mają poprawne maski binarne {0,1} z niezerowym foreground (23/23) oraz typem `uint8` (23/23). Odsetek wokseli urazu był skrajnie niski (min 0.0029%, max 0.0388%, średnio 0.0126%).

4.4.6 Voxel-wise macierz pomyłek (FN vs FP)

Agregowana voxel-wise macierz pomyłek (23 foldy LOOCV) wskazuje na skrajnie niską wykrywalność klasy urazu: TPR = 0.18% (TP = 427, FN = 233.7k), przy bardzo wysokiej swoistości TNR = 99.9977% (FP = 44.8k, TN $\approx 1.93 \times 10^9$). Precyzja wynosi 0.94%, a balanced accuracy ≈ 0.501. Accuracy pomijamy (zdominowane przez

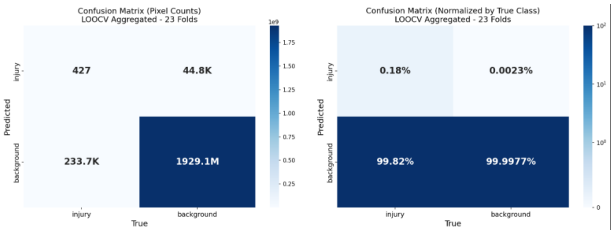


Rysunek 10: Audyt oversamplingu foreground w nnU-Net. Liczba zapamiętanych lokalizacji wokseli klasy urazu (`class_locations`) po preprocessingu dla 23 pacjentów. Linia kropkowana oznacza limit implementacyjny nnU-Net (10 000), a linia przerywana próg minimalny użyteczny dla stabilnego próbkowania. Wszystkie przypadki posiadają niepuste i liczne lokalizacje foreground.



Rysunek 11: Audyt wartości etykiet masek po preprocessingu. Dla każdego pacjenta przedstawiono procent wokseli klasy urazu (foreground) w masce GT. Wszystkie przypadki zawierają wyłącznie wartości {0,1} i niepusty foreground. Linia przerywana oznacza średni udział foreground (0.0126%), wskazując na ekstremalną nierównowagę klas.

TN). Wynik jest spójny z rozkładem Dice o medianie 0 i dominacją przypadków bez overlap.



Rysunek 12: Voxel-wise macierz pomyłek dla segmentacji 3D (nnU-Net) w protokole LOOCV ($n = 23$ foldy). Po lewej surowe zliczenia wokseli (TP, FP, FN, TN) zsumowane po foldach. Po prawej macierz znormalizowana względem klasy prawdziwej (kolumny sumują się do 100%), co uwidacznia TPR = 0.18% dla klasy injury oraz TNR = 99.9977% dla background. Zastosowano skalę logarytmiczną dla czytelności przy silnej nierównowadze klas.

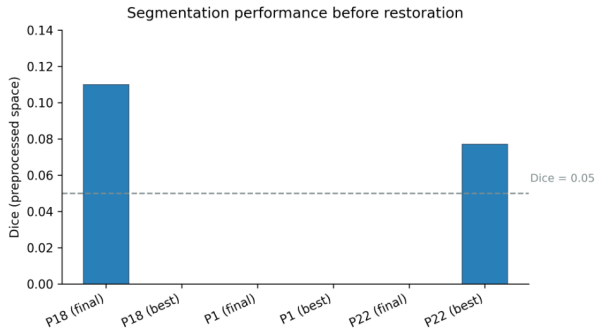
4.4.7 Dice w przestrzeni preprocessingu (bez restoration)

Wykonano inferencję i obliczono Dice bezpośrednio w przestrzeni preprocessingu, raportując liczność wokseli foreground w GT i predykcji.

Dla sześciu uruchomień obliczono Dice bezpośrednio w przestrzeni preprocessingu. W 4/6 uruchomień uzyskano Dice = 0 mimo niepustych predykcji, co **silnie wskazuje**, że brak pokrycia może występować już w przestrzeni treningowej (a więc nie jest to wyłącznie artefakt restoration/resamplingu). Zaobserwowano różnice między checkpointami (`best` vs `final`), w których `best` bywał gorszy od `final`, co sugeruje niestabilną konwergencję lub rozjazd metryki walidacyjnej z jakością segmentacji.

Pacjent	Fold	Checkpoint	Dice (preproc)	|GT| (vox)	|Pred|
18	9	`checkpoint_final.pth`	0.110316	6554	
18	9	`checkpoint_best.pth`	0.000000	6554	
1	0	`checkpoint_final.pth`	0.000000	14053	
1	0	`checkpoint_best.pth`	0.000000	14053	3
22	14	`checkpoint_final.pth`	0.000000	10489	
22	14	`checkpoint_best.pth`	0.076709	10489	

Tabela 7: Sanity-check: Dice w przestrzeni preprocessingu dla wybranych uruchomień (`best` vs `final`). W wielu przypadkach predykcja zawiera woksele foreground, ale brak jest pokrycia z GT (Dice = 0).



Rysunek 13: Wyniki sanity-checku w przestrzeni preprocessingu (bez restoration/resamplingu). Dice obliczono w tej samej przestrzeni, w której trenowano nnU-Net. Dominacja przypadków z Dice = 0 przy niepustych predykcjach wskazuje na problem uczenia/pipeline'u niezależny od etapu przywracania geometrii.

5 Dyskusja

Kontrast pomiędzy wynikiem single-case overfit a LOOCV/LOSO sugeruje, że dominującym ograniczeniem jest reżim małego N oraz wysoka zmienność zmian urazowych między pacjentami, a nie fundamentalna awaria narzędziowa. Test przeuczenia należy traktować jako kontrolę poprawności procesu (pipeline validation), a nie miarę skuteczności klinicznej ani dowód generalizacji.

Możliwy bias selekcyjny i "occult/subtle injuries". Zgodnie z informacją kliniczną uzyskaną podczas przygotowania adnotacji (radiolog, komunikacja osobista, 2025) kohorta mogła być wzbogacona o przypadki wymagające wielokrotnych badań kontrolnych oraz o zmiany subtelne, trudne do uchwycenia w standardowym odczycie. Taki dobór przypadków mógłby pogłębiać efekt małego N i heterogeniczności, ograniczając możliwość uczenia się stabilnych cech między pacjentami.

Hierarchia hipotez dla *collapse* w 2D. Obserwowany spadek metryk po przejściu na split patient-level interpretujemy w następującej kolejności:

(H1) **Artefakt metryk przez przeciek pacjentowy (główny czynnik w złym splicie).** Split per-slice umożliwia, aby niemal identyczne sąsiednie przekroje tego samego pacjenta trafiały do treningu i walidacji, co prowadzi do pozornie wysokich wyników.

(H2) **Ograniczenia danych po naprawie splitu.** Po eliminacji leaku model jest dodatkowo ograniczany przez (i) małe N niezależnych pacjentów oraz (ii) niedoskonałości adnotacji 2D (bbox per-slice), w tym zmienność widoczności urazu na sąsiednich przekrojach i wąskie marginesy, co może wprowadzać *label noise* i utrudniać uczenie stabilnych cech.

Test *lock inference* wskazuje, że raportowane wyniki segmentacji nie są artefaktem niestabilnej ewaluacji ani ustawień inferencji (w tym TTA). W konsekwencji dalsza analiza powinna koncentrować się na przyczynach związanych z uczeniem i danymi (np. ekstremalna nierównowaga foreground, jakość etykiet, sampling, heterogeniczność urazów).

Audyt `class_locations` wyklucza hipotezę, że model nie miał dostępnych wokseli foreground do oversamplingu po preprocessingu; nie przesądza jednak o optymalności oversamplingu ani o tym, czy trening powinien się udać.

Audyt wartości etykiet ($\{0,1\}$, `uint8`, niepusty foreground) wyklucza częsty, "banalny" powód Dice ≈ 0 związany z błędnym formatem masek i wzmacnia interpretację w kategoriach reżimu małego N oraz ekstremalnej nierównowagi klas.

Voxel-wise macierz pomyłek pokazuje, że porażka nie wynika z "zalewu" FP, lecz z prawie całkowitego zaniku czułości na uraz (dominacja FN), co jest typowym trybem awarii przy małym N i silnej nierównowadze wokseli.

6 Wnioski

W reżimie małego N (tutaj: $N = 23$ pacjentów) największym ryzykiem nie jest samo przeuczenie, lecz pozornie bardzo dobre wyniki wynikające z błędnej walidacji. Eksperyment split per-slice w danych CT silnie skorelowanych w osi Z pokazuje, że można uzyskać wysokie metryki jako artefakt przecieku pacjentowego, a nie rzeczywistej zdolności modelu do uogólniania.

Po przejściu na walidację patient-level obraz detekcji ulega dramatycznej zmianie: skuteczność spada, ujawniając dominację błędów FN i FP oraz brak transferu wiedzy między pacjentami (zob. Tabela 5).

Dla segmentacji 3D w protokole LOSO generalizacja praktycznie zanika (zob. Tabela 6). Jednocześnie test single-case overfit potwierdza, że pipeline i etykiety są treningowalne, natomiast głównym ograniczeniem jest brak uogólniania w reżimie małego N i wysoka heterogeniczność między pacjentami.

W praktyce oznacza to, że zadanie wymaga większej liczby niezależnych pacjentów i większej różnorodności zmian, ponieważ sygnał jest subtelny, a korelacje między przekrojami silne.

Zestaw sanity-check (deterministyczność inferencji w lock

inference, audyt `class_locations`, audyt wartości etykiet {0,1} oraz ewaluacja Dice w przestrzeni preprocessingu) eliminuje najczęstsze tryby awarii technicznej i pozwala odróżnić błąd pipeline'u od realnego braku generalizacji. Analiza voxel-wise wskazuje ponadto na porażkę o charakterze utraty czułości na klasę urazu przy bardzo wysokiej swoistości tła (szczegóły: Rys. 12).

Praktyczny wniosek jest następujący: minimalnym standardem raportowania dla małych zbiorów CT powinny być patient-level split, krzywe PR dla detekcji oraz zestaw sanity-check obejmujący kontrolę geometrii, weryfikację metryk, test przeuczenia i audyt przestrzeni preprocessingu. Z perspektywy dalszych prac, wyniki sugerują priorytet strategii data-centric i transferu (pretraining/self-supervised, fine-tuning na zbiorach pokrewnych, podejścia dwuetapowe detekcja→segmentacja) zamiast dalszego strojenia architektury trenowanej od zera.

7 Reproducibility i materiały dodatkowe

Sprzęt, ˙srodowisko i wersje (minimum).

- GPU: NVIDIA GeForce RTX 4080.
- PyTorch: 2.8.0+cu128.
- Ultralytics: 8.4.8.
- nnU-Net: v2.

Zakres raportowania metod i wyników porządkujemy względem checklisty CLAIM (aktualizacja 2024) oraz ogólnych zaleceń raportowania modeli predykcyjnych TRIPOD+AI, aby ułatwić ocenę ryzyka biasu i odtwarzalność eksperymentów.[9, 10]

7.1 Detekcja 2D (YOLO): ustawienia kluczowe

Model: `yolo11m.pt`. Parametry treningu: `imgsz`=512, `batch`=16, `epochs`=100.

Szczegółowe tabele hiperparametrów, wag strat, augmentacji, seedów i dodatkowych ustawień (audit trail) przeniesiono do Aneksu: Sekcja B.

7.2 Segmentacja 3D (nnU-Net): ustawienia kluczowe

Framework: nnU-Net v2, konfiguracja `3d_fullres` (zadanie binarne).

Komendy odtwarzające (LOOCV/LOSO) podano w repozytorium/materiałach dodatkowych wraz z artefaktami audytu i sanity-check.

Etyka, zgody, anonimizacja, dostępność danych i kodu

Etyka i zgody. Ze względu na kliniczny charakter danych (CT) ich przetwarzanie odbywało się zgodnie z obowiązującymi procedurami instytucji ˙zródłowej oraz przepisami dot. ochrony danych. (W razie potrzeby: tutaj należy doprecyzować status zgody komisji bioetycznej/IRB oraz ewentualne zwolnienie z obowiązku uzyskania indywidualnej zgody pacjentów.)

Anonimizacja. Dane obrazowe (DICOM) zostały zanonimizowane przed analizą; usunięto/zanonimizowano identyfikatory pacjenta oraz metadane umożliwiające identyfikację.

Dostępność danych. Dane nie są publicznie dostępne ze względu na ograniczenia prawne i etyczne związane z danymi klinicznymi. Dostęp może być rozważony w uzasadnionych przypadkach na wniosek, po spełnieniu wymogów instytucji ˙zródłowej oraz podpisaniu odpowiednich umów (np. NDA/DTA), o ile pozwalają na to lokalne regulacje.

Dostępność kodu. Skrypty umożliwiające odtworzenie analizy i generowanie wyników/rysunków są dostępne w materiałach uzupełniających (Supplementary Materials).

A Protokół audytu (sanity-check)

W tej sekcji zbieramy minimalny zestaw testów diagnostycznych pozwalających odróżnić błąd pipeline'u (np. geometria, maski, ewaluacja) od rzeczywistego braku generalizacji.

Check	Cel / oczekiwany wynik
Single-case overfit	Model powinien osiągnąć wysoki Dice na jednym przypadku (kontrola treningowalności danych i pipeline'u).
Lock inference (TTA off)	Powtórzona inferencja powinna dawać identyczne wyniki (kontrola deterministyczności i konfiguracji).
Audyt `class_locations`	`class_locations` nie powinno być puste dla przypadków z foreground (kontrola oversamplingu).
Audyt wartości masek {0,1}	Maski powinny być binarne (lub zgodne z założeniami), bez nieoczekiwanych etykiet.
Dice w przestrzeni preprocessingu	Sprawdzenie overlap bez etapu restoration/resamplingu (kontrola, czy problem nie wynika z geometrii).

Tabela 8: Checklista audytu (sanity-check) użyta w pracy.

B Reproducibility detekcji 2D (YOLO): szczegóły (materiały dodatkowe)

B.1 Środowisko i model

Parametr	Wartość
Ultralytics version	8.4.8
Bazowy checkpoint	`yolo11m.pt` (YOLOv11 Medium)
Framework	PyTorch (przez Ultralytics)

Tabela 9: Środowisko i model dla detekcji 2D (YOLO).

B.2 Hiperparametry treningu

Parametr	Wartość	Uwagi
Input size (`imgsz`)	512×512	
Batch size	16	
Epochs	100	
Optimizer	auto	AdamW dla małych batch, SGD dla dużych
Initial LR (`lr0`)	0.01	
Final LR factor (`lrf`)	0.01	końcowe LR = `lr0` × `lrf`
Momentum	0.937	
Weight decay	0.0005	
Warmup epochs	3.0	
Warmup momentum	0.8	
Warmup bias LR	0.1	
Scheduler	Linear decay	cosine, jeśli `cos_lr=True`

Tabela 10: Hiperparametry treningu YOLO użyte w eksperymentach detekcji 2D.

B.3 Wagi składowych funkcji straty (custom)

Parametr	Wartość	Opis
`box`	7.5	Waga `box loss`
`cls`	0.5	Waga `classification loss`
`dfl`	1.5	Waga `distribution focal loss`

Tabela 11: Wagi funkcji straty w YOLO (ustawienia custom).

B.4 Augmentacje

Augmentacja	Wartość	Status / opis
`mosaic`	0.0	WYŁĄCZONE (domyślnie 1.0)
`mixup`	0.0	WYŁĄCZONE
`copy_paste`	0.0	WYŁĄCZONE
`hsv_h`	0.015	Hue shift ±1.5%
`hsv_s`	0.7	Saturation ±70%
`hsv_v`	0.4	Value ±40%
`translate`	0.1	±10%
`scale`	0.5	±50%
`fliplr`	0.5	50% chance
`flipud`	0.0	Wyłączone
`erasing`	0.4	40% chance
`degrees`	0.0	Bez rotacji
`shear`	0.0	Bez shear
`perspective`	0.0	Bez perspektywy

Tabela 12: Augmentacje YOLO użyte w treningu detekcji 2D.

B.5 Seed i powtarzalność

Parametr	Wartość
`seed`	0
`deterministic`	True
Liczba uruchomień	1 run per fold (23 foldy LOSO)

Tabela 13: Ustawienia powtarzalności dla detekcji 2D (YOLO).

B.6 Dodatkowe ustawienia: binary vs multi-class

Poniżej rozdzielamy ustawienia dla dwóch trybów detekcji: (i) binarnego (wszystkie klasy scalone) oraz (ii) wieloklasowego.

Parametr	Wartość
`single_cls`	True (wszystkie klasy jako jedna)
`close_mosaic`	0 (bez zamykania mosaic przed końcem)

Tabela 14: Dodatkowe ustawienia YOLO: tryb binarny (`single_cls=True`).

Binary (`single_cls=True`).

Parametr	Wartość
`single_cls`	False (klasy trenowane osobno)
`close_mosaic`	0 (bez zamykania mosaic przed końcem)

Tabela 15: Dodatkowe ustawienia YOLO: tryb wieloklasowy (`single_cls=False`).

Multi-class (`single_cls=False`). Ablacje (detekcja 2D). Wyniki ablacji (rozmiar modelu, augmentacje, downsampling) podsumowano w Sekcji Wyniki.

Czas obliczeń. Średni czas treningu na fold: ~ 18.5 h, łącznie ~ 425 h (~ 18 dni) dla protokołu LOSO ($n = 23$ foldy).

C Szczegółowe statystyki zbioru (materiały dodatkowe)

W głównym tekście przedstawiono wyłącznie liczby podsumowujące (Tabela 2). Poniżej zamieszczamy rozkłady per pacjent oraz rozkład zasięgu przekrojów (slice) przypadających na pojedynczą instancję urazu.

C.1 Rozkłady per pacjent

Ze względu na małe N oraz ryzyko nadinterpretacji wyników na poziomie pojedynczych przypadków (a także ograniczenia anonimizacji), nie zamieszczamy w publikacji tabeli z rozkładami per pacjent.

C.2 Rozkład zasięgu slice per uraz

Przez *zasięg slice per uraz* rozumiemy liczbę kolejnych przekrojów z, na których dana instancja urazu jest obecna w masce 3D (np. długość przedziału $[z_{\min}, z_{\max}]$ dla tej instancji).

W niniejszej wersji pracy pomijamy tabelę kwartylową zasięgu slice na uraz, ponieważ byłaby ona nadmiernie szczegółowa w relacji do głównych wniosków (a puste miejsca obniżają czytelność raportu).

D Preset windowingu (materiały dodatkowe)

W pracy stosujemy standardowy windowing CT; w wariancie 2.5D używamy trzech **stałych** presetów okien jako trzech kanałów wejściowych (multi-window). Aby uniknąć *target leakage*, presety te są niezależne od klasy urazu.

D.1 Presety 3-kanałowe (wariant RSNA 2019)

Wariant "RSNA 2019" odnosi się do ustawień stosowanych w ramach RSNA 2019 Brain CT Hemorrhage Challenge.[11]

Wartości okien (C/W) użyte w podejściu wielokanałowym (3 kanały) inspirowane ustawieniami stosowanymi w zadaniach krwawień śródczaszkowych:

Preset	C	W
mózg (brain)	40	80
podtwardówkowe (subdural)	80	200
kości (bone)	600	2800

Tabela 16: Stałe presety windowingu (C/W) stosowane jako kanały wejściowe w wariancie 2.5D (multi-window).

D.2 Uwaga kliniczna (zmienność i kalibracja)

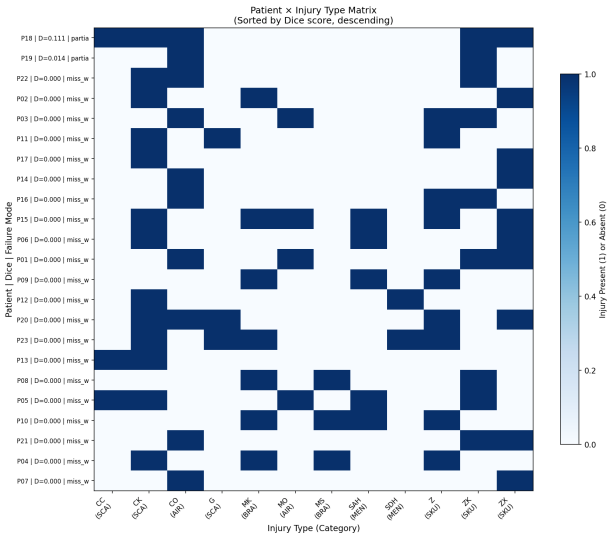
Zwracamy uwagę, że preferencje windowingu mogą istotnie zależeć od warunków oglądania (monitor, oświetlenie) oraz celu diagnostycznego (liczba jednocześnie ocenianych struktur). Jednym z potencjalnych kierunków rozwoju jest rozszerzenie wejścia do 5–7 kanałów oraz kalibracja okien na reprezentatywnych obszarach tkanek, z jawnym zdefiniowaniem "punktów styku" dla różnych patologii.

E Augmentacje nnU-Net (materiały dodatkowe)

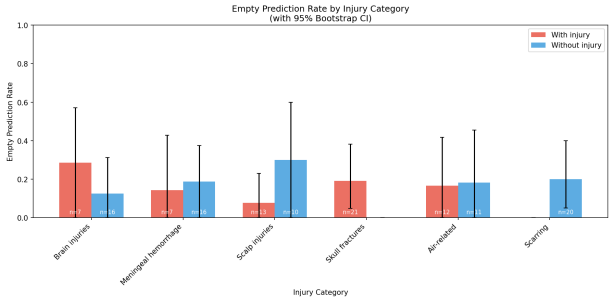
Poniżej zestawiono skrótowy opis augmentacji 3D stosowanych w nnU-Net (wariant domyślny). Parametry te traktujemy jako element reprodukowalności (audit trail), a nie jako strojenie modelu.

- Rotacje: do $\pm 15^\circ$ wokół osi.
- Skalowanie: $[0.85,\ 1.25]$.
- Elastic deformation: włączone.
- Mirroring: osie (x, y, z).
- Gamma: $[0.7,\ 1.5]$, $p = 0.15$ (retain stats).
- Additive brightness, Gaussian noise, Gaussian blur, simulate low resolution.
- `do_dummy_2d_data_aug`: False.

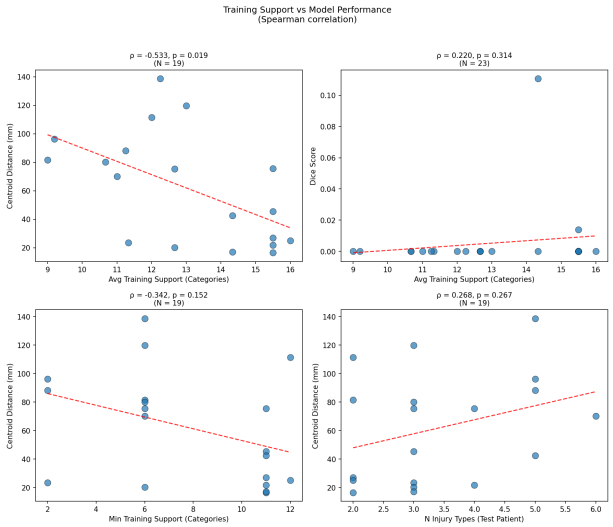
F Failure mode analysis i rzadkość urazów (materiały dodatkowe)



Rysunek 14: Macierz pacjent × typ urazu (1 = obecny, 0 = brak), posortowana wg Dice (malejąco). Widoczna jest wysoka heterogeniczność kombinacji urazów oraz koncentracja niezerowych Dice w pojedynczych przypadkach.



Rysunek 15: Rysunek D2: Odsetek pustych predykcji z 95% bootstrap CI dla przypadków z daną kategorią urazu oraz bez niej. Ze względu na małe N i nierówne grupy wyniki traktujemy jako eksploracyjne.



Rysunek 16: Rysunek D3: Zależność miar wsparcia w treningu od jakości predykcji: (i) centroid distance (N=19, tylko niepuste predykcje), (ii) Dice (N=23). Wyniki dla dystansu są obciążone biasem selekcji, ponieważ przypadki z pustą predykcją są wykluczone.

Literatura

[1] Ultralytics, *YOLOv8* (software repository), accessed 2026-01-28.

[2] E. Yagis, S. W. Atnafu, A. García Seco de Herrera, et al., *Effect of data leakage in brain MRI classification using 2D convolutional neural networks*, Scientific Reports, 2021;11:22544. doi:10.1038/s41598-021-01681-w.

[3] T. J. Bradshaw, Z. Huemann, J. Hu, and A. Rahmim, *A Guide to Cross-Validation for Artificial Intelligence in Medical Imaging*, Radiology: Artificial Intelligence, 2023;5(4):e220232. doi:10.1148/ryai.220232.

[4] S. Varma and R. Simon, *Bias in error estimation when using cross-validation for model selection*, BMC Bioinformatics, 2006;7:91. doi:10.1186/1471-2105-7-91.

[5] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, *nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation*, Nature Methods, 2021;18(2):203–211. doi:10.1038/s41592-020-01008-z.

[6] S. Ostmeier, B. Axelrod, J. Bertels, et al., *USE-Evaluator: Performance metrics for medical image segmentation models supervised by uncertain, small or empty reference annotations in neuroimaging*, Medical Image Analysis, 2023;90:102927. doi:10.1016/j.media.2023.102927.

[7] A. A. Taha and A. Hanbury, *Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool*, BMC Medical Imaging, 2015;15:29. doi:10.1186/s12880-015-0068-x.

[8] D. P. Chakraborty, *A brief history of free-response receiver operating characteristic paradigm data analysis*, Academic Radiology, 2013;20(7):915–919. doi:10.1016/j.acra.2013.03.001.

[9] A. S. Tejani, M. E. Klontzas, A. A. Gatti, et al., *Checklist for Artificial Intelligence in Medical Imaging (CLAIM): 2024 Update*, Radiology: Artificial Intelligence, 2024;6(4):e240300. doi:10.1148/ryai.240300.

[10] *TRIPOD+AI statement: updated guidance for reporting clinical prediction model studies that use machine learning* (TRIPOD+AI).

[11] A. E. Flanders, L. M. Prevedello, G. Shih, et al., *Construction of a Machine Learning Dataset through Collaboration: The RSNA 2019 Brain CT Hemorrhage Challenge*, Radiology: Artificial Intelligence, 2020;2(3):e190211. doi:10.1148/ryai.2020190211.

[12] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, *Tversky loss function for image segmentation using 3D fully convolutional deep networks*, 2017.