

Raport

Sieci Neuronowe w Histopatologii

Anomaly Detection

Wojciech Kosiuk
Politechnika Warszawska

Szymon Matuszewski
Politechnika Warszawska

Michał Mazuryk
Politechnika Warszawska

Tomasz Modzelewski
Politechnika Warszawska

Tymoteusz Urban
Politechnika Warszawska

4 czerwca 2023

Spis treści

1	Wstęp	2
2	Algorytmy PO Ekstrakcji Cech	2
2.1	Analiza podejścia	2
2.1.1	Plan działania	2
2.1.2	Rak jako outlier	2
2.1.3	Model boosting	2
2.2	Algorytmy Anomaly Detection	3
2.3	Wyniki dla zdjęć RGB	3
2.3.1	Model Sigmoid oraz ImageNet	3
2.3.2	Model Boosting	3
2.3.3	Model z Warstwy Konwolucyjnej	4
2.3.4	Model z Autoencoderem	4
2.4	Zdjęcia Czarnobiałe	6
2.4.1	Model Sigmoid oraz ImageNet	6
2.4.2	Model Boosting	7
2.4.3	Model z Warstwy Konwolucyjnej	7
2.4.4	Model z Autoencoderem	7
2.5	Wnioski	8
3	Algorytmy PRZED Ekstrakcją Cech	8
3.1	Analiza podejścia	8
3.1.1	Model f-AnoGAN	8
3.1.2	Normalizacja plam przy użyciu StainNet	9
3.1.3	Przycinanie zdjęć	9
3.2	Generowanie	9
3.3	Wyniki	10
3.3.1	Przykładowy przebieg uczenia	10
3.3.2	Wyniki na zbiorze testowym	10
3.3.3	Klasyfikacja przy użyciu anomaly score	11
3.4	Wnioski	13
4	Podsumowanie	13
5	Literatura	13

1 Wstęp

Poniższa praca stanowi raport przygotowany na Warsztaty Badawcze prowadzone na Wydziale Matematyki i Nauk Informacyjnych Politechniki Warszawskiej w roku akademickim 2022/23 w semestrze letnim. Zagadnienie nad jakim pracowaliśmy to Anomaly Detection na zdjęciach histopatologicznych. Biorąc pod uwagę podejście medyczne anomalią przyjętą przez nas jest komórka rakowa. W celu ich wykrycia zastosowaliśmy dwa podejścia:

1. Algorytmy PO ekstrakcji cech ze zdjęcia,
2. Algorytmy PRZED ekstrakcją cech ze zdjęcia.

2 Algorytmy PO Ekstrakcji Cech

2.1 Analiza podejścia

2.1.1 Plan działania

Aby użyć zdjęć histopatologicznych (konkretnie patchy) do algorytmów Anomaly Detection dostosowanych do danych tabularycznych potrzebowaliśmy najpierw zmniejszyć rozmiar danych. Postanowiliśmy stworzyć model autoenkodera, który pozwoli zakodować informacje za pomocą mniejszego wektora, który użyjemy jako reprezentację patchy do AD. Zdecydowaliśmy się również na wytrenowanie modeli z warstwami konwolucyjnymi i głębokimi. Przetwarzając przez nie patch będziemy uzyskiwać z modelu ostatnią warstwę konwolucyjną - czyli informację po zmniejszeniu wielkości obrazu, przedostatnią warstwę głęboką - czyli informację w pewien sposób przetworzoną przez model. Warstwy te będziemy traktować jako wektory cech, do algorytmów AD. W zależności od wyciągniętej warstwy będziemy sprawdzać inne podejście badawcze.

2.1.2 Rak jako outlier

Uzyskując warstwę konwolucyjną oraz zakodowaną z autoenkodera, będziemy chcieli sprawdzić czy algorytmy są w stanie na optymalnym poziomie klasyfikować anomalie jako raka. Stosunek patchy z rakiem w porównaniu ze zdrowymi patchami w zbiorze musieliśmy ustawić na kilka procent i taki sam dać jako argument do algorytmów AD, aby mogły one klasyfikować je jako anomalie.

2.1.3 Model boosting

Za model boosting rozumiemy podejście próbujące poprawić decyzyjność wytrenowanego modelu. Nasza hipoteza badawcza przy tym podjęciu sprawdzała czy pewne grupy patchy, z którymi model ma problem (błędą predykce) mogą różnić się zakodowaną informacją od tych, które przewiduje dobrze. Za zakodowaną informację będzie nam służyć wyciągnięty wektor cech z przedostatniej warstwy modelu, czyli ten, który jest użyty do podjęcia ostatecznej decyzji w modelu.

Zgadzałoby się to z logiką uczenia, ponieważ dany element lub kombinacja elementów w tak wyjętym wektorze powinna odpowiadać za znalezienie poszczególnych informacji biologicznych w patchu. Reasumując, pewne patche, na których model się myli, zawierające specyficzne kombinacje biologiczne, mogłyby być znalezione przez algorytmy AD, ponieważ zakodowany wektor ich cech powinien się różnić od reszty.

Znajdywanie takich problematycznych patchy mogłoby posłużyć do douczania modelu lub analizy przez histopatologów. Przy wylapywaniu wysokiego odsetka nawet do bezpośredniego zwiększenia skuteczności modelu - przykładowo jeśli model miałby 85% accuracy, a AD oznaczałaby 15% zbioru jako anomalię, z której 80% byłoby znalezionymi błędami modelu, to ustawiając odwrotną predykcję wszędzie gdzie AD wykrył anomalię otrzymalibyśmy: $0.85 + 0.15 \cdot 0.8 - 0.15 \cdot 0.2 = 0.94 = 94\%$ accuracy takiego dwuetapowego przejścia, stąd użyta nazwa model boostingu.

Sample number	Ground Truth	Model prediction	Anomaly algorithm prediction
1	1	1	0
2	1	0	1
3	0	0	0
4	0	1	1

Tabela 1: Każdy wiersz to pewien patch poddany predykcji. W taki sposób działałby idealny AD przy tym podejściu, czyli wyłapywałby wszystkie patche, dla których model się myli.

2.2 Algorytmy Anomaly Detection

Traktując outliery w zbiorze danych jako próbki posiadające komórki rakowe, mogliśmy wykorzystać algorytmy Anomaly Detection do wykrywania komórek rakowych. Temat ten jest znany w medycynie i algorytmy te są nierzadko wykorzystywane w danych medycznych. Natomiast, znaleźliśmy dosyć niewiele artykułów korzystających z algorytmów anomaly detection na danych histopatologicznych. W naszej pracy wykorzystaliśmy 3 algorytmy do wykrywania anomalii w zbiorze danych: Isolation Forest, Local Outlier Factor, One-Class SVM. Dodatkowo, jako 4 model użyty został model głoszący na podstawie decyzji modeli bazowych. Poniżej została przedstawiona krótka charakterystyka i budowa każdego z trzech algorytmów:

- Isolation Forest - opiera się na wykorzystaniu losowych drzew decyzyjnych do izolowania rzadkich obserwacji, umożliwiając identyfikację anomalii poprzez ich szybkie oddzielenie od większości danych.
- Local Outlier Forest - ocenia stopień odosobnienia punktów danych na podstawie porównania ich gęstości lokalnej z gęstością ich sąsiadów, umożliwiając w ten sposób identyfikację obserwacji nietypowych lub odstających od większości danych.
- One-Class Support Vector Machine (One-Class SVM) - konstruuje granicę decyzyjną wokół jednej klasy danych, a następnie identyfikuje punkty odstające, które znajdują się poza tą granicą

2.3 Wyniki dla zdjęć RGB

2.3.1 Model Sigmoid oraz ImageNet

Stworzony przez nas został model sekwencyjny sieci neuronowej, który składał się z kilku warstw, Conv2D z 32 filtrami o rozmiarze 3x3 i funkcją aktywacji ReLU, następnie stosowana była warstwa MaxPooling2D z oknem o rozmiarze 2x2, aby zmniejszyć rozmiar danych. Następnie warstwa Conv2D z 64 filtrami i ponownie warstwą MaxPooling2D. Po tym następuje kolejno dodawane są dwie warstwy dense z funkcją aktywacji ReLU, z 128 i 30 neuronami odpowiednio, na końcu funkcja aktywacji sigmoidalnej. Model jest kompilowany z optymalizatorem 'adam' i funkcją straty 'binary crossentropy'. Był on uczony na próbce 50/50 patchy z rakiem do zdrowych.

Oprócz własnego modelu postanowiliśmy skorzystać z modelu przedtrenowanego na zbiorze zdjęć ImageNet VGG16 i dotrenować go na danych histopatologicznych. Model ten składa się z 16 warstw głębokich, z czego 13 warstw konwolucyjnych oraz 5 warstw MaxPooling. Do jego wyjścia dodaliśmy jedną warstwę głęboką o liczbie neuronów 128. Jego rezultaty zostały przedstawione w tabeli poniżej:

	Sigmoid	ImageNet
Accuracy	0.76	0.75
Precision	0.71	0.83
Recall	0.89	0.62
F1 Score	0.79	0.72

2.3.2 Model Boosting

Dla podejścia poprawy jakości modeli przetwarzając ich przedostatnie warstwy, procent outlierów do dostosowania algorytmów był ustawiony na procent błędnych predykcji na zbiorze testowym w

modelach, czyli dla sigmoid na 24%, a dla modelu image-net na 25%. Wyniki są przedstawione dla zbioru walidacyjnego (nie użytego wcześniej):

	Isolation Forest	Local Outlier Factor	One-Class SVM	Voting
Accuracy	0.58	0.66	0.55	0.60
Precision	0.15	0.30	0.07	0.12
Recall	0.15	0.33	0.07	0.10

Tabela 2: Wyniki dla modelu sigmoid

	Isolation Forest	Local Outlier Factor	One-Class SVM	Voting
Accuracy	0.68	0.61	0.72	0.69
Precision	0.11	0.30	0.51	0.41
Recall	0.24	0.30	0.06	0.27

Tabela 3: Wyniki dla modelu VGG16

2.3.3 Model z Warstwy Konwolucyjnej

Jako alternatywne podejście zastosowaliśmy przejście przez warstwy konwolucyjne modelu i użycie outputu z ostatniej warstwy konwolucyjnej jako wejścia do algorytmów na danych tabularycznych do wykrywania anomalii opisanych wcześniej. Zbiór użyty do dostosowania algorytmów anomalii miał 15% patchy z rakiem, więc parametr ilości anomalii również wynosił 15%. Ich wyniki na zbiorze walidacyjnym prezentują się następująco:

	Isolation Forest	Local Outlier Factor	One-Class SVM	Voting
Accuracy	0.70	0.68	0.71	0.74
Precision	0.04	0.04	0.15	0.07
Recall	0.04	0.04	0.16	0.05

Tabela 4: Wyniki dla modelu sigmoid

2.3.4 Model z Autoencoderem

Kolejną próbą poprawy jakości modelu było użycie autoenkodera. Klasycznie, autoenkodery stosowane są do kompresji i rekonstrukcji obrazów. Skorzystaliśmy z pierwszej funkcjonalności i przeprowadziliśmy próby wyciągnięcia najistotniejszych cech i użycia reprezentacji wektorowej zdjęć RGB o wymiarach 96x96 do algorytmów Anomaly Detection. Kodowanie było realizowane poprzez warstwy konwolucyjne, warstwy poolingowe i jedną warstwę dense. Model został wytrenowany przy użyciu optymalizatora Adam i funkcji straty MSE. Tak przygotowane dane tabularyczne zostały wykorzystane do algorytmów anomaly detection. Ich rezultaty na zbiorze walidacyjnym zostały przedstawione w tabeli poniżej:

	Isolation Forest	Local Outlier Factor	One-Class SVM	Voting
Accuracy	0.69	0.67	0.70	0.72
Precision	0.04	0.02	0.07	0.03
Recall	0.04	0.02	0.06	0.02

Tabela 5: Wyniki dla modelu imagenet

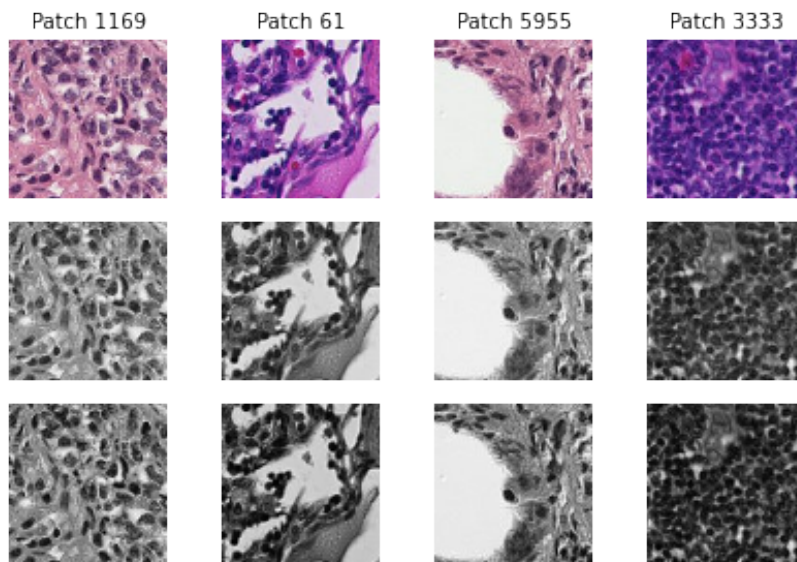
	Isolation Forest	Local Outlier Factor	One-Class SVM	Voting
Accuracy	0.70	0.68	0.67	0.72
Precision	0.21	0.12	0.11	0.16
Recall	0.17	0.10	0.08	0.09

2.4 Zdjęcia Czarnobiałe

Postanowiliśmy zastosować identyczne podejście również do zdjęć w skali szarości. Chcieliśmy sprawdzić hipotezę, czy anomalie w powyższych algorytmach wykrywane są w dużej zależności od koloru tkanki. Niektóre źródła, takie jak [?], sugerują na poprawę jakości modeli klasyfikujących zdjęcia medyczne po transformacji tych zdjęć na czarnobiałe.

Kroki, które poczyniliśmy w celu transformacji zdjęć histopatologicznych w skalę szarości:

1. Zamiana zdjęć **RGB** na **grayscale** (skala szarości).
2. Transformacja **CLAHE** zdjęć **grayscale**.
3. Rozszerzenie wymiaru kolorów poprzez powtórzenie czarnobiałego wymiaru - *repeat expand dims*, w celu imitacji obrazu RGB wymaganego przez algorytmy.



Rysunek 1: Na rysunku kolejne etapy transformacji: wiersz górny - zdjęcia RGB, wiersz środkowy - grayscale, wiersz dolny - CLAHE. Zdjęcia do prezentacji zostały wylosowane.

Efekty transformacji możemy zauważyć na rysunku 1. Jak widzimy transformacja Adaptive Histogram Equalization nie wprowadza drastycznych zmian dla oka, jednak może mieć wpływ na jakość modeli.

2.4.1 Model Sigmoid oraz ImageNet

Model sigmoid, tak jak dla zdjęć RGB był obliczany dla proporcji 0.5 komórek zdrowych do wszystkich komórek. Chcieliśmy, aby porównanie względem zdjęć kolorowych było sprawiedliwe. ImageNet z kolei, analogicznie był tworzony dla proporcji 0.85. Wyniki na zbiorze walidacyjnym prezentujemy w tabeli 6. Oba modele nie były w stanie nauczyć się wykrywania anomalii. Transformacje w tym przypadku nie przynosiły zadowalających rezultatów.

	Sigmoid	ImageNet
Accuracy	0.50	0.85
Precision	0	0
Recall	0	0
F1 Score	0	0

Tabela 6: Tabela przedstawia wyniki modeli na zdjęciach po transformacji grayscale oraz CLAHE. Liczba epoch obydwu modeli wynosi 25.

2.4.2 Model Boosting

W celu sprawdzenia algorytmów Anomaly Detection poprawiających jakość wykrywania modelu przeprowadziliśmy analogiczne rozumowanie jak w przypadku zdjęć RGB. Wyniki przedstawione są w tabeli 7. W porównaniu do rozwiązania ze zdjęciami kolorowymi nie daje to pozytywnych rezultatów.

	Isolation Forest	Local Outlier Factor	One-Class SVM	Voting
Accuracy	0.67	0.68	0.66	0.68
Precision	0.12	0.17	0.12	0.14
Recall	0.20	0.29	0.19	0.20

Tabela 7: Tabela przedstawia wyniki algorytmów Anomaly Detection mających na celu poprawić jakość wykrywania komórek rakowych na zbiorze walidacyjnym.

2.4.3 Model z Warstwy Konwolucyjnej

Zastosowanie algorytmów Anomaly Detection na ostatniej warstwie konwolucyjnej również nie przyniosło zadowalających efektów. Wyniki przedstawione są na tabeli 8.

	Isolation Forest	Local Outlier Factor	One-Class SVM	Voting
Accuracy	0.81	0.80	0.82	0.90
Precision	0.01	0.01	0.03	0.02
Recall	0.05	0.18	0.09	0.05

Tabela 8: Tabela przedstawia wyniki algorytmów wykrywających anomalie z ostatniej warstwy konwolucyjnej na zbiorze walidacyjnym. Jak widzimy algorytmy te nie były w stanie wykrywać komórek rakowych.

2.4.4 Model z Autoencoderem

Analogiczne rozumowanie w przypadku zastosowania Autoencodera przeprowadziliśmy w przypadku zdjęć czarnobiałych. Wytrenowany Autoencoder posiadał SSIM Score na poziomie 0.75, więc możemy go uznać za całkiem reprezentatywny.

	Isolation Forest	Local Outlier Factor	One-Class SVM	Voting
Accuracy	0.74	0.71	0.72	0.76
Precision	0.16	0.05	0.12	0.14
Recall	0.17	0.05	0.13	0.12

Tabela 9: Tabela przedstawia wyniki modeli Anomaly Detection po zastosowaniu autoencodera na zdjęciach po transformacjach grayscale oraz CLAHE.

2.5 Wnioski

- Wyniki z danych tabularycznych są tak dobre jak modele - im lepszy autoenkoder tym bardziej szczegółowe cechy zakodowane, im lepszy model tym mniejszy procent jego błędów co pozwala na ustawienie parametru outlierów do algorytmów AD na mniejsze wartości (przy ustawieniu parametru na 24% algorytmy nie działają tak dobrze jak dla rzędu kilku %)
- Algorytmy AD odpowiadają za uczenie nienadzorowane przez co nie możemy wskazać w jaki sposób patche chcielibyśmy żeby były klasyfikowane. Mogą one znajdować inne cechy odstające nie tak jak chcieliśmy w zadaniu.
- Dla wielkich danych nie są efektywne przy dostosowaniu (fit), ustawienie pojedynczego algorytmu to czas rzędu godziny, z powodu czasu oczekiwania, jak i słabych wyników ograniczyliśmy się do prezentowania wyników dla najlepiej nauczonych algorytmów (bez statystycznego podawania niepewności)
- Transformacja zdjęć na czarnobiałe nie wspomogła algorytmów Anomaly Detection w wykrywaniu komórek rakowych. Kolor zdjęcia (komórki) jest wykorzystywany przez te algorytmy w określeniu anomalii.

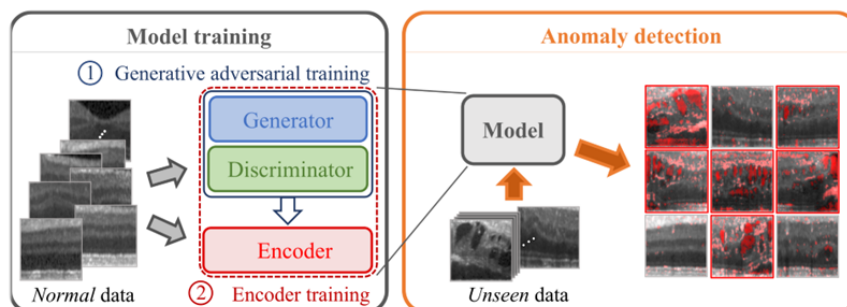
3 Algorytmy PRZED Ekstrakcją Cech

3.1 Analiza podejścia

W tej części badaliśmy algorytmy przed ekstrakcją cech ze zdjęcia. Tak jak poprzednio, za anomalię uznaliśmy komórkę rakową. Do ich wykrywania użyliśmy sieci typu generative adversarial network (GAN). Sprawdziliśmy również, jaki wpływ na osiągnięte rezultaty ma zastosowanie techniki normalizacji plam (stain normalization).

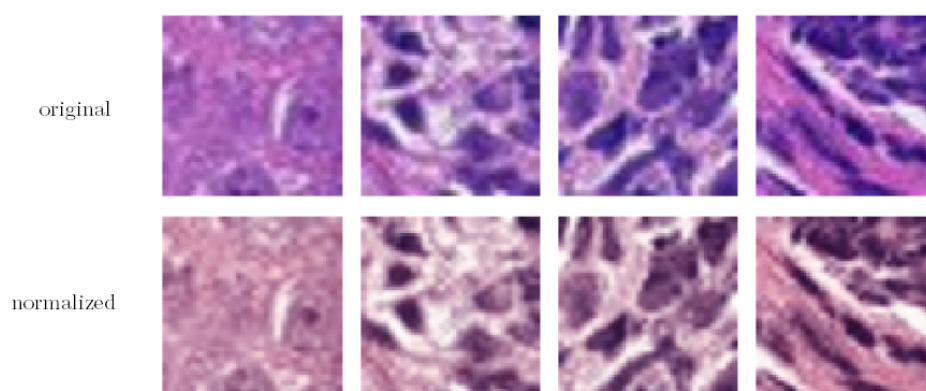
3.1.1 Model **f-AnoGAN**

Schemat modelu **f-AnoGAN** przedstawiono na rysunku poniżej. Dane wejściowe w postaci zdjęcia najpierw są kodowane przez autoenkoder do postaci wektora ukrytego. Następnie następuje proces generowania zdjęcia z tak otrzymanego wektora przy użyciu generatora. Współczynnik *anomaly score* obliczany jest jako miara różnicy pomiędzy zdjęciem wejściowym a otrzymanym. Im bardziej te zdjęcia się różnią, tym większe jest prawdopodobieństwo, że mamy do czynienia z anomalią. Uczenie odbywa się na danych normalnych, a więc w naszym przypadku na zdjęciach z etykietą 0, czyli zdrowych. W pierwszej kolejności uczona jest sieć GAN (składająca się z generatora i dyskryminatora), a następnie - autoenkoder. Proces uczenia przeprowadzono na zdjęciach ze zbioru treningowego PCAM (domyślnie 262144 elementów, ale ze względów wydajnościowych uczenie przeprowadzono na mniejszych pozbiórach). Następnie na zbiorze testowym sprawdzamy, jak prezentują się wartości współczynnika *anomaly score* przy próbkach pozytywnych i negatywnych, tj. czy model prawidłowo wykrywa anomalie.



3.1.2 Normalizacja plam przy użyciu StainNet

Aby polepszyć wyniki algorytmu, wykonaliśmy także normalizację plam przy użyciu sieci StainNet. Skorzystaliśmy z wytrenowanej sieci udostępnionej przez autorów modelu¹. Poniżej przedstawiono efekty normalizacji na przykładowych zdjęciach zbioru PCAM.



3.1.3 Przycinanie zdjęć

Zastosowaliśmy również transformację polegającą na przycięciu oryginalnego zdjęcia o wymiarach 96×96 i pozostawieniu środka o wymiarach 32×32 , albowiem dodatnia etykieta implikuje, że w środku tym znajdują się komórki rakowe.

3.2 Generowanie

Proces uczenia przeprowadzono 5 razy według następujących podejść:

1. Default (30): GAN - 30 epok, encoder - 20 epok
2. Cropped (15): GAN - 15 epok, encoder - 15 epok, z przycinaniem
3. Cropped (7): GAN - 7 epok, encoder - 10 epok, z przycinaniem
4. Validation (50): GAN - 50 epok, encoder - 20 epok, z przycinaniem
5. StainNet (10): GAN - 10 epok, encoder - 10 epok, z przycinaniem i normalizacją plam

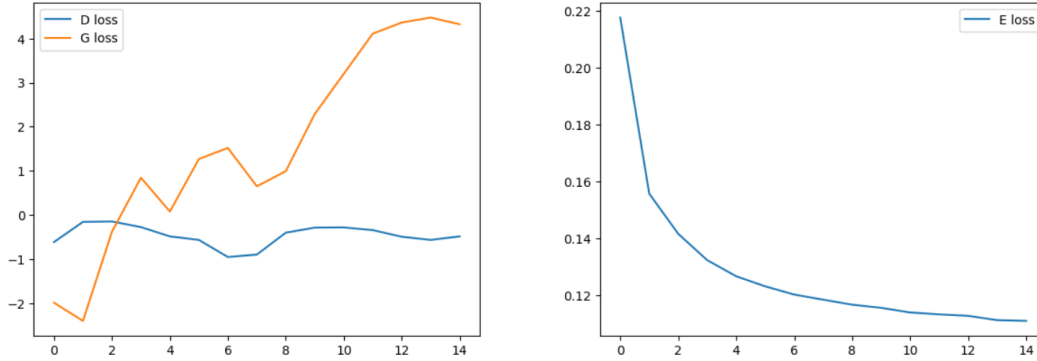
Ze względów wydajnościowych, proces uczenia przeprowadzono na podzbiorze składającym się z losowo wybranej 1/4 elementów spośród zdjęć ze zbioru uczącego o etykiecie 0 (zdrowe). Wyniki były każdorazowo obliczane na podstawie 32768 próbek ze zbioru testowego (W wersji Validation wyniki były testowane na zbiorze walidacyjnym).

¹https://github.com/khtao/StainNet/blob/master/checkpoints/camelyon16_dataset/StainNet-Public-centerUni.layer3_ch32.pth

3.3 Wyniki

3.3.1 Przykładowy przebieg uczenia

Poniżej przedstawiono wartości funkcji straty sieci GAN oraz autoenkodera po poszczególnych epokach (dla uczenia opisanego w 3.2.3, tj. po 15 epok z przycinaniem zdjęć).



Na lewym wykresie widać, że istnieje problem z dobrym wytrenowaniem generatora, gdyż wartości funkcji straty ciągle rosną, a powinny od pewnego momentu maleć. Dzieje się tak niezależnie czy liczba epok wynosi 15 czy 50. Widzimy jednak, że trening encodera przebiegał pomyślnie.

3.3.2 Wyniki na zbiorze testowym

Wartości metryki *anomaly score* na zbiorze testowym dla poszczególnych generatorów przedstawiają się następująco:

	anomaly score (mean)	anomaly score (median)	anomaly score (max)
normal	0.121854	0.125204	1.686141
cancer	0.131875	0.126825	0.419204

Tabela 10: Wyniki na zbiorze testowym - Default (30)

	anomaly score (mean)	anomaly score (median)	anomaly score (max)
normal	0.099689	0.100745	0.746071
cancer	0.112805	0.103960	0.542333

Tabela 11: Wyniki na zbiorze testowym - Cropped (15)

	anomaly score (mean)	anomaly score (median)	anomaly score (max)
normal	0.106780	0.104006	0.483460
cancer	0.104986	0.098597	0.448402

Tabela 12: Wyniki na zbiorze testowym - Cropped (7)

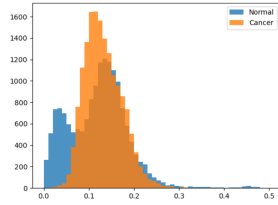
	anomaly score (mean)	anomaly score (median)	anomaly score (max)
normal	0.107628	0.105604	1.281195
cancer	0.107537	0.097927	0.380977

Tabela 13: Wyniki na zbiorze testowym - Validation (50)

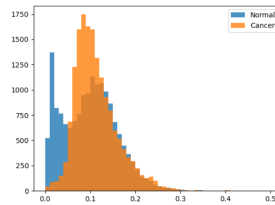
	anomaly score (mean)	anomaly score (median)	anomaly score (max)
normal	0.107470	0.111191	0.382624
cancer	0.105713	0.101831	0.351290

Tabela 14: Wyniki na zbiorze testowym - StainNet (10)

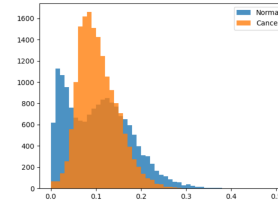
Poniższe histogramy przedstawiają wartości współczynnika anomaly score dla poszczególnych generatorów.



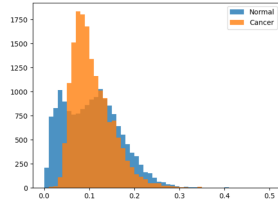
Default (30)



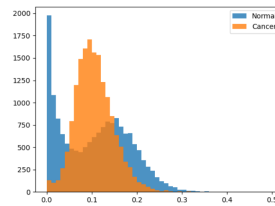
Cropped (15)



Cropped (7)



Validation (50)

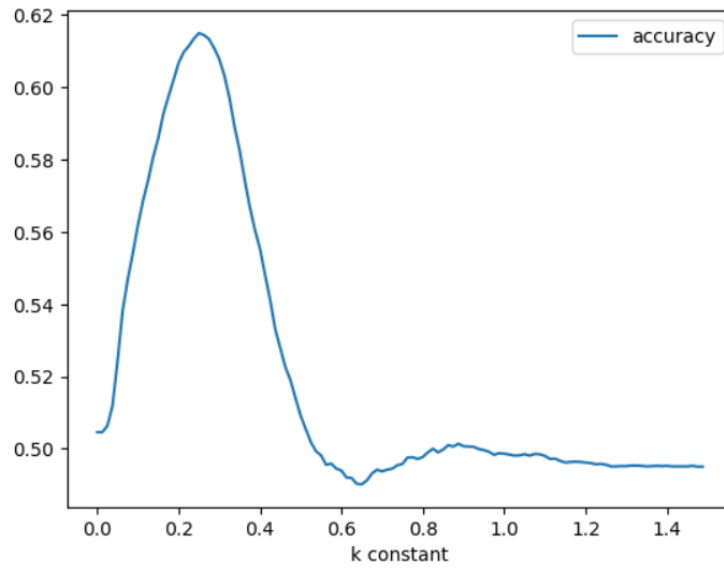


StainNet (10)

3.3.3 Klasyfikacja przy użyciu anomaly score

Naszym celem jest rozstrzygnięcie (na podstawie uzyskanych wyników anomaly score), czy dany obiekt jest anomalią. Oczwistym podejściem będzie więc próba znalezienia takiego punktu odcięcia (cut-off point), powyżej którego zdjęcie będzie klasyfikowane jako anomalia. Punkt ten wyznaczymy ze wzoru $k \cdot (a_0 + a_1)$, gdzie a_0 i a_1 to średnie wartości anomaly score na zbiorze treningowym odpowiednio dla zdjęć zdrowych i chorych, zaś k - pewna ustalona stała.

Poniższy wykres przedstawia poszczególne wartości accuracy score na zbiorze testowym w zależności od współczynnika k (dla uczenia z punktu 3.2.3).



Optymalna wartość współczynnika w tym przypadku to $k = 0.25$. Analogiczne obliczenia przeprowadziliśmy dla każdego z pozostałych czterech generatorów; średni wynik to $k = 0.2375$. Ostatecznie więc klasyfikator rozpoznaje zdjęcie jako anomalię wtedy i tylko wtedy, gdy anomaly score tego zdjęcia jest większe od $0.2375(a_0 + a_1)$, gdzie a_0 i a_1 to średnie wartości anomaly score odpowiednio dla zdjęć zdrowych i chorych na zbiorze, na którym odbywało się uczenie danej sieci.

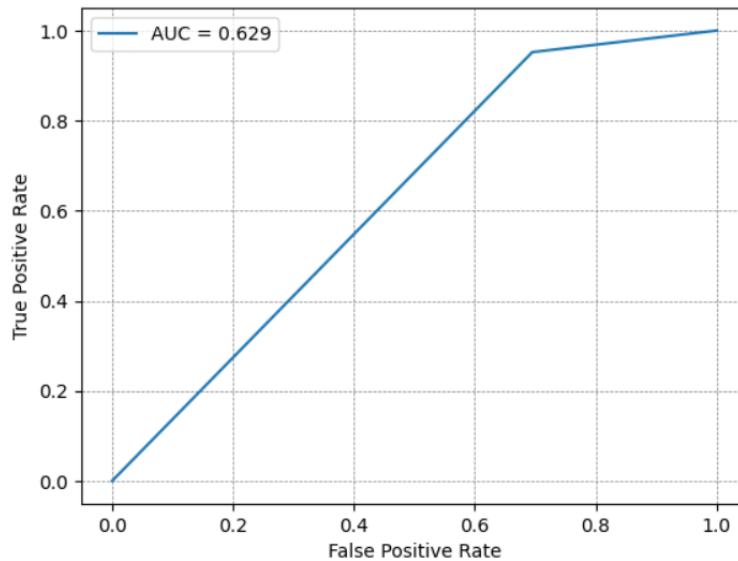
Poniższa tabela przedstawia wyniki dla poszczególnych generatorów ($k = 0.2375$).

	Accuracy	Precision	Recall	f1-score
Default (30)	0.607339	0.563206	0.988207	0.717493
Cropped (15)	0.613366	0.569439	0.958422	0.714415
Cropped (7)	0.615426	0.572885	0.934684	0.710371
Validation (50)	0.594797	0.557312	0.959039	0.704961
StainNet (10)	0.630836	0.583829	0.934533	0.718679

Najlepsze wyniki udało się uzyskać dla danych po przycinaniu i normalizacji (StainNet). Okazuje się, że wyniki możemy nieco poprawić, jeżeli zastosujemy współczynnik k optymalny dla StainNet ($k = 0.2125$):

	Accuracy	Precision	Recall	f1-score
StainNet (10)	0.631522	0.582516	0.952071	0.722796

Poniższy wykres przedstawia krzywą ROC dla tego ostatniego klasyfikatora:



3.4 Wnioski

Algorytmy wykrywające anomalie, działające na zdjęciach przed ekstrakcją cech i oparte na sieci f-AnoGAN, charakteryzowały się wysoką pełnią i niską precyzją. Innymi słowy, niemal wszystkie przypadki raka zostaną rozpoznane, jednak często będziemy uzyskiwać wyniki fałszywie pozytywne. Najlepsze rezultaty uzyskaliśmy po uprzednim przycięciu zdjęcia do środka o wymiarach 32×32 , z jednoczesną normalizacją plam przy użyciu sieci StainNet.

4 Podsumowanie

Podsumowując, testowane przez nas rozwiązania nie do końca dobrze się sprawdziły w przypadku wykrywania raka. Widzimy co prawda duży potencjał w rozwiązaniu f-AnoGAN, jednakże do pełnego wykorzystania tej technologii potrzebne są czas i ogromne zasoby obliczeniowe, by móc wytrenować model na dużej liczbie danych i wystarczająco dużej liczbie epok.

5 Literatura

- 'Learning image representations for anomaly detection: application to discovery of histological alterations in drug development' - Igor Zingman, Birgit Stierstorfer
- Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U., 2019. f-AnoGAN: Fast Unsupervised Anomaly Detection with Generative Adversarial Networks. Medical Image Analysis 54, 30-44. DOI: <https://doi.org/10.1016/j.media.2019.01.010>
- Implementation of f-AnoGAN with PyTorch, <https://github.com/A03ki/f-AnoGAN>
- Kang H, Luo D, Feng W, Zeng S, Quan T, Hu J and Liu X (2021) StainNet: A Fast and Robust Stain Normalization Network. Front. Med. 8:746307. doi: 10.3389/fmed.2021.746307

6 Podział pracy

- Za część algorytmów po ekstrakcji cech odpowiadali: Michał Mazuryk, Wojciech Kosiuk, Szymon Matuszewski
- Za część algorytmów przed ekstrakcją cech odpowiadali: Tymoteusz Urban, Tomasz Modzelewski