



# **Zastosowanie auto-encoderów wariacyjnych do rozpoznawania zmian na obrazach medycznych**

(English title)

Tomasz Nanowski

Praca inżynierska

**Promotor:** dr Jan Chorowski

Uniwersytet Wrocławski  
Wydział Matematyki i Informatyki  
Instytut Informatyki

24 stycznia 2019

Tomasz Nanowski

.....

.....

(adres zameldowania)

.....

.....

(adres korespondencyjny)

PESEL: .....

e-mail: .....

Wydział Matematyki i Informatyki

stacjonarne studia I stopnia

kierunek: informatyka

nr albumu: 279076

### **Oświadczenie o autorskim wykonaniu pracy dyplomowej**

Niniejszym oświadczam, że złożoną do oceny pracę zatytułowaną *Zastosowanie auto-encoderów wariacyjnych do rozpoznawania zmian na obrazach medycznych* wykonałem/am samodzielnie pod kierunkiem promotora, dr. Jana Chorowskiego. Oświadczam, że powyższe dane są zgodne ze stanem faktycznym i znane mi są przepisy ustawy z dn. 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (tekst jednolity: Dz. U. z 2006 r. nr 90, poz. 637, z późniejszymi zmianami) oraz że treść pracy dyplomowej przedstawionej do obrony, zawarta na przekazanym nośniku elektronicznym, jest identyczna z jej wersją drukowaną.

Wrocław, 24 stycznia 2019

(czytelny podpis)



## Streszczenie

Celem projektu jest wykorzystanie modelu VAE do wykrywania zmian patologicznych w obrazach medycznych

---

...



# Spis treści

<b>1. Introduction</b>	<b>11</b>
<b>2. Artificial neural networks</b>	<b>13</b>
2.1. Autoencoders . . . . .	13
2.2. VAE . . . . .	13
2.3. Convolutional VAE . . . . .	14
2.4. Deep feature consistent variational auto-encoder . . . . .	14
<b>3. Experiments on MNIST</b>	<b>15</b>
3.1. MNIST . . . . .	15
3.2. VAE . . . . .	15
3.2.1. ROC . . . . .	17
3.3. Deep feature consistent variational auto-encoder . . . . .	17
<b>4. Experiments on MRI FLAIR images</b>	<b>21</b>
4.1. Data description & preprocessing . . . . .	21
4.1.1. Overview . . . . .	21
4.1.2. Patches . . . . .	22
4.1.3. Preprocessing . . . . .	23
4.1.4. Normalizations . . . . .	24
4.2. Results . . . . .	24
4.3. Analysis . . . . .	24
4.4. Back to basic models . . . . .	26
<b>5. Workflow</b>	<b>29</b>

**6. Summary****31**



# Notes

■ Napisać w końcu opis VAE . . . . .	13
■ Uzupełnić podpisy . . . . .	15
■ Do sprawdzenia . . . . .	17
■ Opisać gdzieś te koszty . . . . .	17
■ Uzupełnić podpisy . . . . .	22
■ Może LIME? . . . . .	24
■ Rozważyć normalizacje: histogramowa, tylko środkowy kanał . . . . .	24
■ dodać krótki opis modeli i może dla czego softmax . . . . .	24
■ Dodać z-dim . . . . .	24



**Rozdział 1.**

**Introduction**



## Rozdział 2.

# Artificial neural networks

Sztuczne sieci neuronowe mają obecnie bardzo mocno ugruntowaną pozycję szczególnie w dziedzinie problemów związanych z analizą i przetwarzaniem obrazów. Pomimo, iż nie jest to nowy pomysł, dopiero ostatni wzrost w wydajności komputerów pozwolił na ich praktyczne zastosowanie. Z matematycznego punktu widzenia są to sparametryzowane nieliniowe funkcje o pewnej ustalonej strukturze. Składają się z prostych elementów zwanych neuronami, a one natomiast są pogrupowane w warstwy. Połączenia między warstwami definiują przepływ danych. 'Nauka' sieci neuronowych polega na optymalizacji pewnej funkcji straty, czyli wyznaczeniu takich parametrów, żeby osiągnąć minimalny koszt. Do tego celu często korzysta się z metod opartych na SGD, a przy wybranej strukturze można w efektywny sposób zastosować algorytm propagacji wstecznej. W dalszej części pracy będę używał prostszej nazwy (neural nets). Przykładowa architektura sieci neuronowych jest zaprezentowana na wykresie ???.

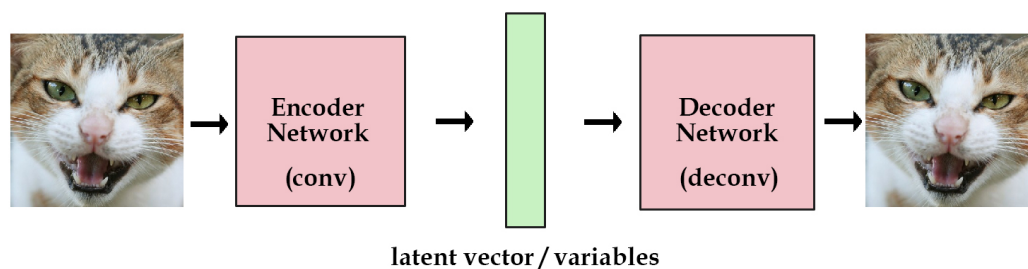
### 2.1. Autoencoders

Jest to jeden z rodzajów sieci neuronowych, służący do znajdowania wydajnej reprezentacji danych, co jest przykładem nauki bez nadzoru. W autoencoder'ach można wyróżnić dwie części: encoder i decoder. Zadaniem encodera jest wyprodukowanie reprezentacji, natomiast decoder służy do odtworzenia z niej oryginalnej postaci. Zależy nam na tym, żeby wyjście było w jakimś sensie jak najbardziej podobne do wejścia. W przypadku obrazów jako funkcja straty często stosowane jest MSE. Przykładowy schemat na wykresie 2.1.

### 2.2. VAE

Variational autoencoders rezszerzają założenia o wprowadzenie modelowania rozkładu prawdopodobieństwa dla reprezentacji ukrytej.

Napisać w końcu opis



Rysunek 2.1: Architecture of autoencoder

### 2.3. Convolutional VAE

Jest to rozszerzenie poprzedniego modelu, w którym dodatkowo stosujemy warstwy splotowe. Szczególnie w przypadku obrazów pozwala to na zwiększenie rozmiaru danych wejściowych przez zmniejszenie ilości parametrów w stosunku do warstw fully-connected oraz wykryciu na wstępie jakiś prostych cech, przez co w reprezentacji mogą znajdować się bardziej abstrakcyjne rzeczy.

### 2.4. Deep feature consistent variational auto-encoder

Ta wersja zakłada użycie innej funkcji kosztu. MSE z samej definicji przyczynia się do uśredniania wartości pikseli przez co wyjściowy obraz nie jest wyraźny. W tym przypadku będziemy korzystać z zewnętrznej sieci splotowej wyuczonej do klasyfikacji obrazów. Będziemy teraz myśleć, że dwa obrazy są podobne, jeśli mają podobne wartości aktywacji w tej sieci. Takie podejście powinno nam dać ostrzejsze wyjście.

## Rozdział 3.

# Experiments on MNIST

### 3.1. MNIST

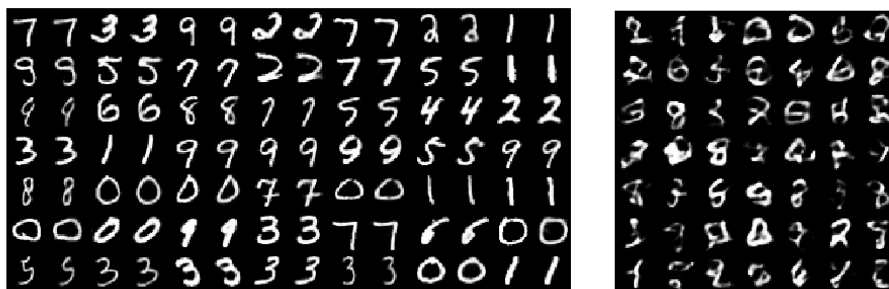
Jest to zbiór po kategoryzowanych odręcznie napisanych cyfr. Wszystkie obrazki są czarno-białe, rozmiaru 28x28 i wycelowane. Zbiór składa się z 60000 danych treningowych i 10000 testowych. Zbiór ten często wykorzystywany jest w celach testowych. W sensie, że jeżeli model na nim nie zadziała, to z dużym prawdopodobieństwem nie zadziała również na bardziej skomplikowanych danych. Przykładowe obrazki 3.1.



Rysunek 3.1: Samples from MNIST dataset

### 3.2. VAE

Na rysunku 3.2 znajduje się efekt wyuczenia modelu VAE z warstwą ukrytą rozmiaru 20. Po lewej widać rekonstrukcje, a po prawej efekty zdekodowania wektora wygenerowanego ze standardowego rozkładu normalnego.

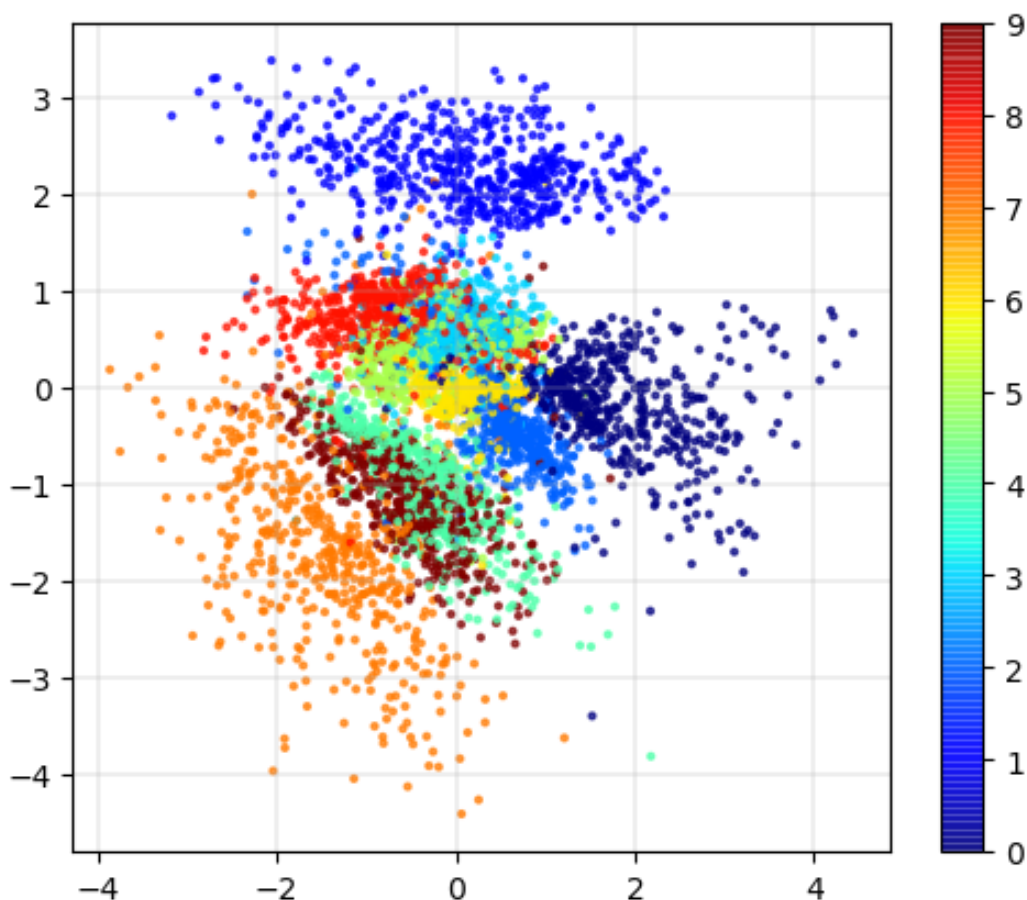


(a) Coffee.

(b) More coffee.

Rysunek 3.2: The same cup of coffee. Two times.

Dodatkowo warto byłoby zobaczyć jak konkretne cyfry rozrzucone są w przestrzeni. 20 wymiarów jest dosyć trudne do zwizualizowania, więc wyuczyłem model dla reprezentacji ukrytej rozmiaru 2. Na rysunku 3.3 znajdują się wyniki. Wartym odnotowania jest fakt, że niektóre klasy są bardzo dobrze separowalne, mimo iż podczas nauki nie staraliśmy się rozwiązywać problemu klasyfikacji. Te ciekawą własność wyniku reprezentacji będę chciał wykorzystać w późniejszej analizie.



Rysunek 3.3



W oryginalnym problemie mamy bardzo duży dysonans pomiędzy ilością przykładów dla każdej z klas. Według statystyk dane z komórkami rakowymi stanowią 2% wszystkich. Cieężko jest więc nawet mówić o jakimś sensownym podejściu supervised. Do zasymulowania tego problemu dla MNIST będę uczył model jedynie na dwóch klasach [4, 7], a następnie testował zachowanie reprezentacji ukrytej dla przykładów z klasy 5.

Do sprawdzenia

Rozmiar reprezentacji ukrytej wynosi 10. Analizować natomiast będziemy 2 składowe koszty dla modelu VAE: KLD oraz błąd rekonsrukcji MSE. Na rysunku 3.4 znajduje się przedstawienie ich wraz z histogramami. Można zauważyć, że wyłącznie na podstawie samego błędu rekonstrukcji można z bardzo dużą dokładnością określić dane pochodzące z klasy 5, mimo iż model nie widział żadnych ich przykładów podczas uczenia.

Opisać gdzieś te koszty



Rysunek 3.4

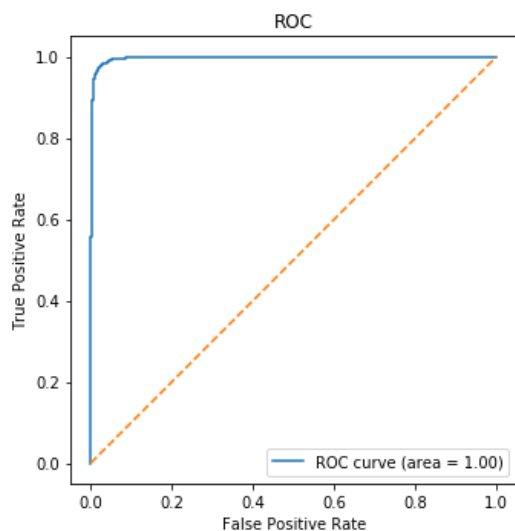
Do określenia jak rzeczywiście dobra jest ta separacja można wykorzystać krzywą ROC. Traktujemy to jako problem binarnej klasyfikacji, gdzie dane z klasy 5 będą oznaczone jako 1, a z [4, 7] jako 0. Wartość confidence to suma kosztów KLD i MSE. Jak widać na rysunku 3.5 takie podejście osiąga prawie 100% skuteczność. Podobne podejście będę chciał zastosować do danych medycznych.

### 3.2.1. ROC

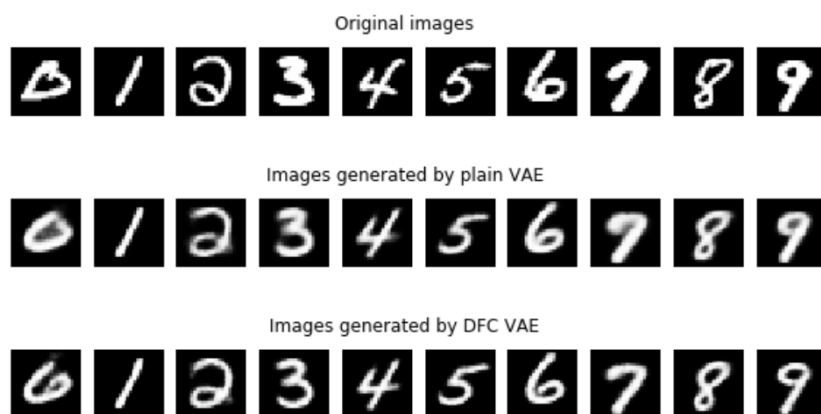
Opis ROC

## 3.3. Deep feature consistent variational auto-encoder

Podobny eksperyment jw. przeprowadziłem również dla modelu DFC. Na początku jednak warto zobaczyć co zyskujemy na zmianie podejścia do kosztu rekonsrukcji. Różnice prezentują się na rysunku 3.6. Widać, że rekonstrukcje są mniej rozmażane niż przy standardowym VAE. Dodatkowo lepiej rekonstruuje takie elementy jak np. pozioma kreska w cyfrze 7.

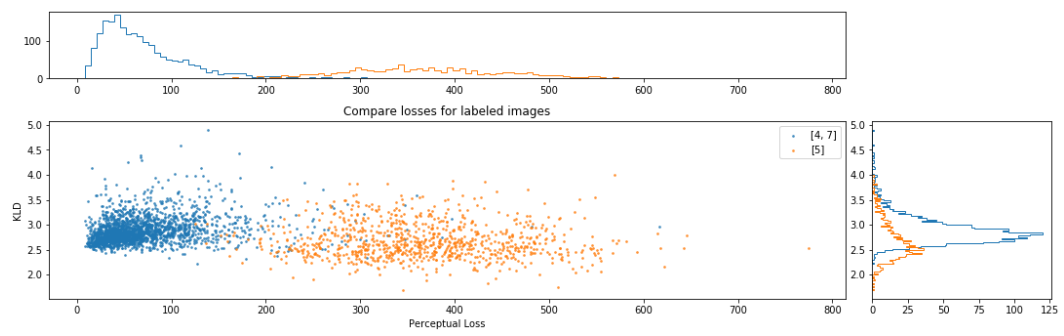


Rysunek 3.5



Rysunek 3.6

Na rysunku 3.7 przedstawiony jest efekt przeprowadzenia analogicznego eksperymentu z nauką na jedynie przykładach z klas [4, 7] i sprawdzeniu zachowania dla danych z klasy 5. Zamiast kosztu rekonstrukcji MSE używamy błędu perceptualnego. Jak widać separacja jest co najmniej tak dobra jak w przypadku zwykłego VAE.



Rysunek 3.7



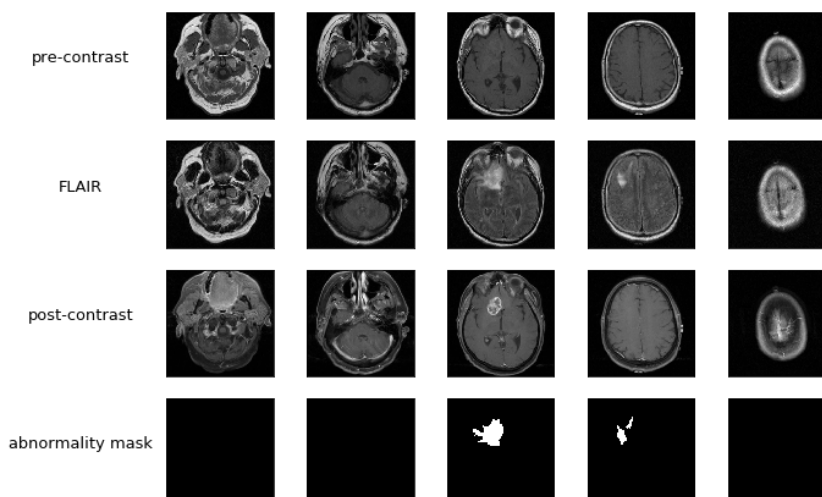
## Rozdział 4.

# Experiments on MRI FLAIR images

### 4.1. Data description & preprocessing

#### 4.1.1. Overview

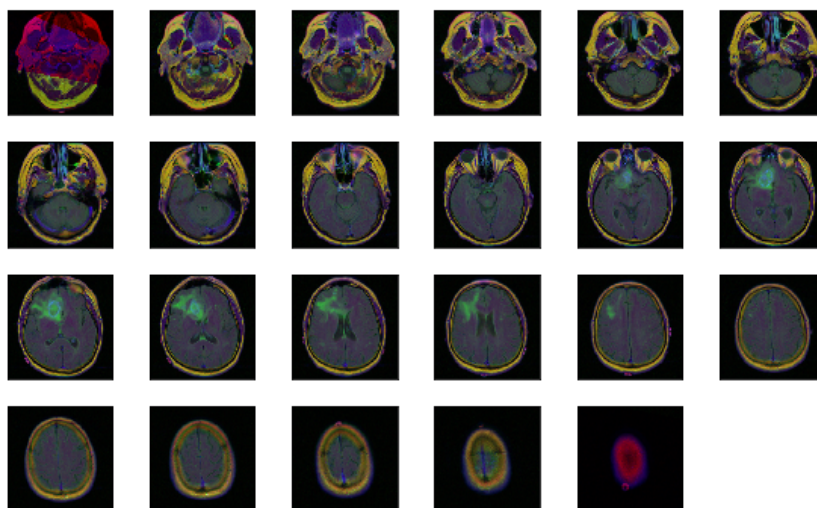
Dane pochodzą z Duke University. Są to zdjęcia z rezonansu magnetycznego głowy wraz z zaznaczonymi obszarami zmian nowotworowych. Obrazy są rozmiaru 256x256x3. Pierwszy kanał jest zdjęciem wykonanym przed wprowadzeniem kontrastu, drugi w trakcie, a trzeci po. Maski mają rozmiar 256x256, a każdy piksel ma wartość 0 albo 255 (255 oznacza komórkę nowotworową). Na rysunku 4.1 pokazane są osobne kanały oraz maska.



Rysunek 4.1

Dane pogrupowane są dla 110 pacjentów. Zestaw dla jednego z nich znajduje się na rysunku 4.2. Łącznie znajduje się 3929 par obrazów, z czego w 1373 jest przynaj-

mniej jedna komórka nowotworowa ( $\sim 34.95\%$ ). Natomiast w całym zbiorze piksele oznaczające zmiany nowotworowe stanowią jedynie  $\sim 1.02988\%$  wszystkich pikseli. Przy tak znaczącej dysproporcji ciężko liczyć na rzetelne podejście supervised, dlatego będę chciał spróbować podejścia unsupervised. Założeniem wykorzystania VAE będzie patrzenie na koszty. Hipotezą jest to, że obrazki z rakiem, czyli takie mało prawdopodobne będą zauważalne w rozkładzie i rekonstrukcji.



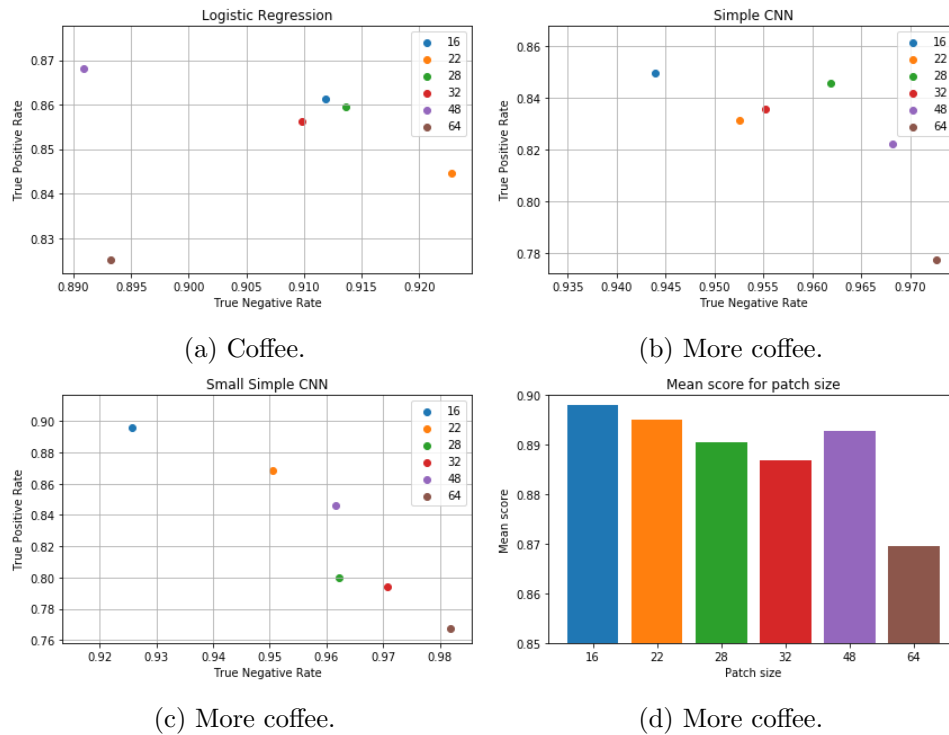
Rysunek 4.2

Pacjenci zostali podzieleni na 2 zbiory: treningowy (70%), testowy (30%). Obrazki oznaczające komórki rakowe będę nazywał pozytywnymi, a te bez negatywnymi.

#### 4.1.2. Patches

Rekonstrukcje nie chcemy robić od razu z całego obrazku, a jedynie z niewielkich wycinków. Rozmiar wycinków ustale na podstawie wyników osiągniętych przez modele supervised, tj. regresja liniowa, małe cnn (mnist), cnn (cifar10). Będę testował rozmiary 16, 22, 28, 32, 48, 64.

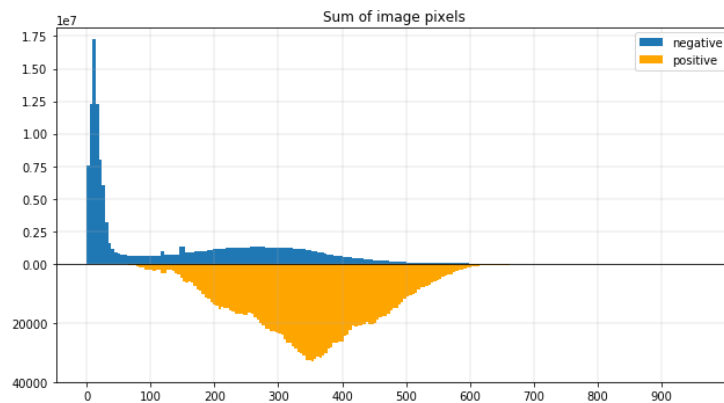
Na obrazku 4.3 widać wyniki dla poszczególnych modeli i rozmiarów. Patrzymy na 2 wartości true positive rate i true negative rate. Najlepszy przypadek jest w takim razie w punkcie (1, 1). Obrazek (d) prezentuje średnie wyniki dla każdego z rozmiarów. Przy takim liczeniu najlepsze wyniki wyszły dla rozmiarów odpowiednio 16 i 22. W dalszych eksperymentach będę korzystał z pacy o rozmiarze 22 ze względu na trochę większe pole obserwacji.



Rysunek 4.3: The same cup of coffee. Two times.

### 4.1.3. Preprocessing

Na obrazku 4.4 przedstawiony jest histogram zsumowanych wartości pikseli dla każdego patchu rozmiaru 22 z podziałem na klasy. Minimalna wartość sumy dla klasy pozytywnej wynosi 56.5. Oznacza to, że ze zbioru treningowego można usunąć obrazki o mniejszej sumie, gdyż automatycznie są to negatywne obrazki. Usunięcie obrazków o wartości mniejszej niż 42 zmniejszy dane o 45.5% i usunięcie trywialnych przypadków czyli prawie całych czarnych obrazków.



Rysunek 4.4

#### 4.1.4. Normalizations

IME?

Rozważyć normalizacje: histogramowa, tylko środkowy kanał

## 4.2. Results

Przetestuję 4 kombinacje modeli, ze zmienioną funkcją kosztu

krótki opis modeli i  
la czego softmax

W tabeli 4.1 zostały przedstawione wyniki dla czterech kombinacji modeli oraz parametrów. Wartości liczbowe opisują najlepsze AUC ROC podczas uczenia wraz z odpowiadającą epoką. Jak widać w tym zestawieniu najlepiej wypadły podstawowe modele VAE.

Model	Best ROC AUC
VAE	0.68 (1)
VAE + Softmax	0.68 (1)
C-VAE	0.59 (1)
C-VAE + Softmax	0.67 (1)

Tablica 4.1: My table

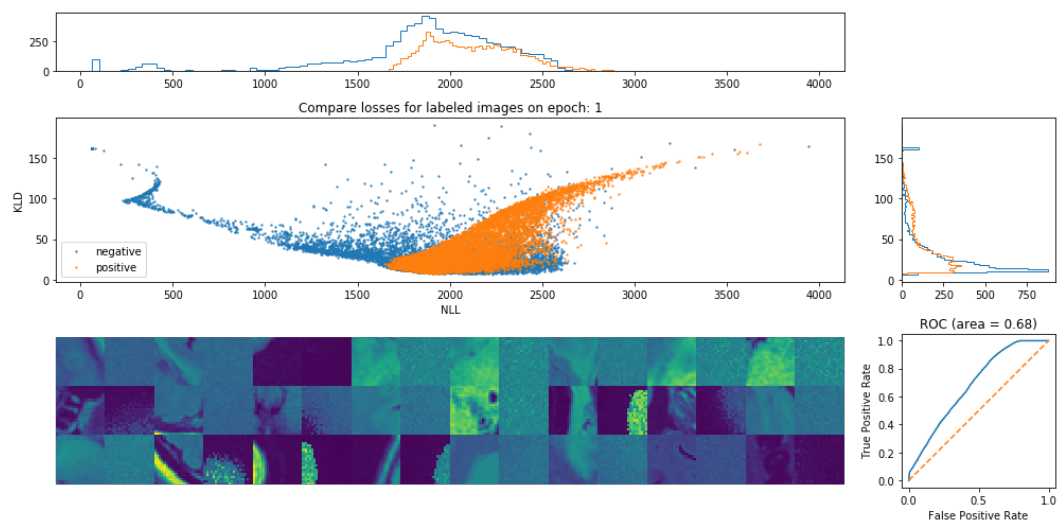
z-dim

## 4.3. Analysis

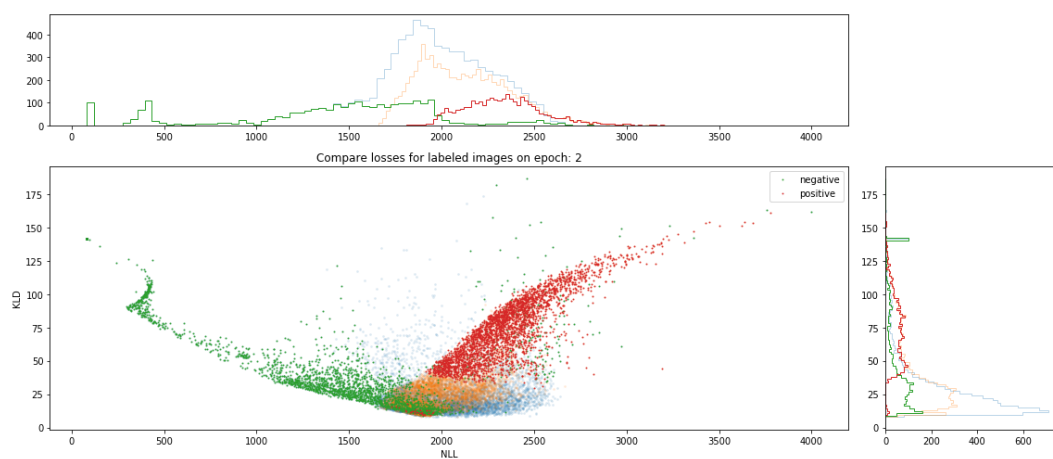
Zajmę się analizą modelu w wersji VAE + Softmax. Na wykresie 4.5 przedstawione są koszty, ich rozkłady, przykłady rekonstrukcji oraz krzywa ROC. Ciekawie wyglądają te dwa zauważalne ogony na wykresie. Postaram się lepiej przyjrzeć ich specyfice.

Pierwszym pomysłem jest, żeby sprawdzić rozkład obrazków w zależności od ich jasności, czyli łącznej sumy pikseli. Ciemne obrazki definiuję jako te o niskiej sumie, a jasne o wysokiej. Na rysunku 4.6 zaznaczyłem negatywne obrazki, dla których suma  $\mu = 50$  oraz pozytywne obrazki z sumą co najmniej 150. Zaznaczone są one odpowiednio zielonym i czerwonym kolorem na wykresie. Jak widać po rozkładach składają się one na dokładnie te 2 separowalne ogony. Wniosek z tego jest następujący. Na podstawie tego modelu możemy odesparować jedynie najciemniejsze i najjaśniejsze obrazki z obydwu klas. W takim razie model nie robi nic nadzwyczajnego. Wystarczy przyjrzeć się histogramowi na rysunku 4.7 z zaznaczonymi progami. One już bardzo dobrze separują te klasy.

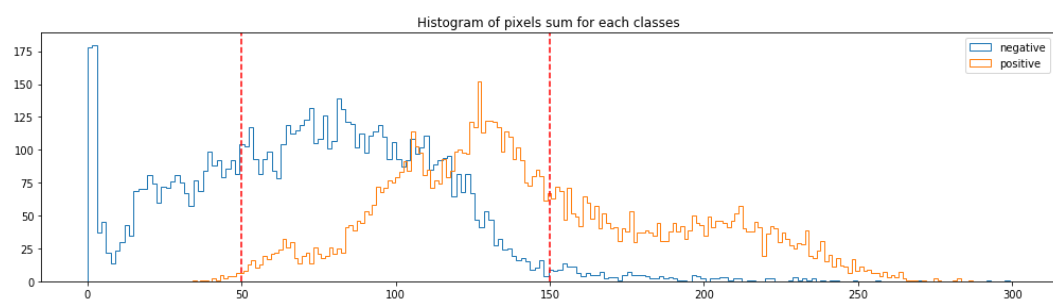




Rysunek 4.5

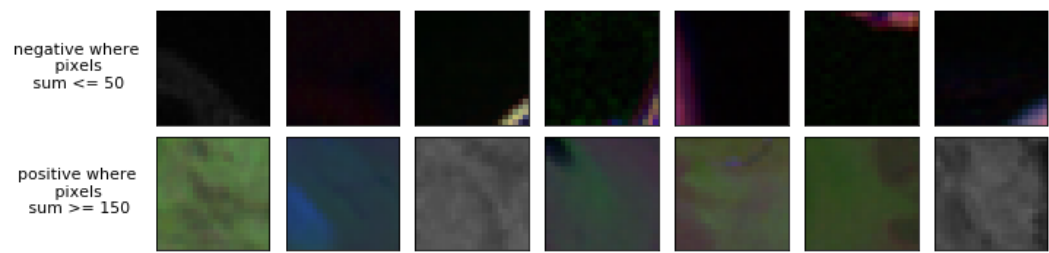


Rysunek 4.6



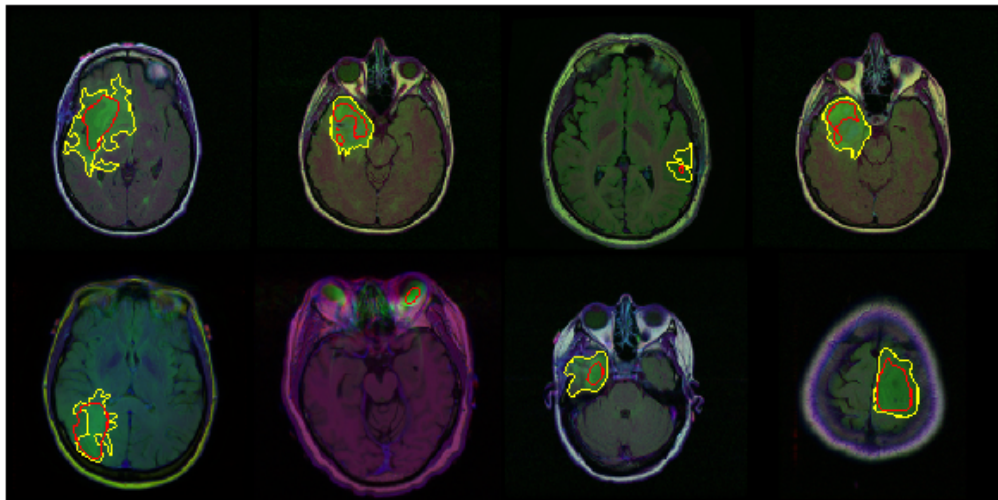
Rysunek 4.7

Na rysunku 4.8 pokazane są przykłady obrazków z zadanymi poziomami sumy pikseli.

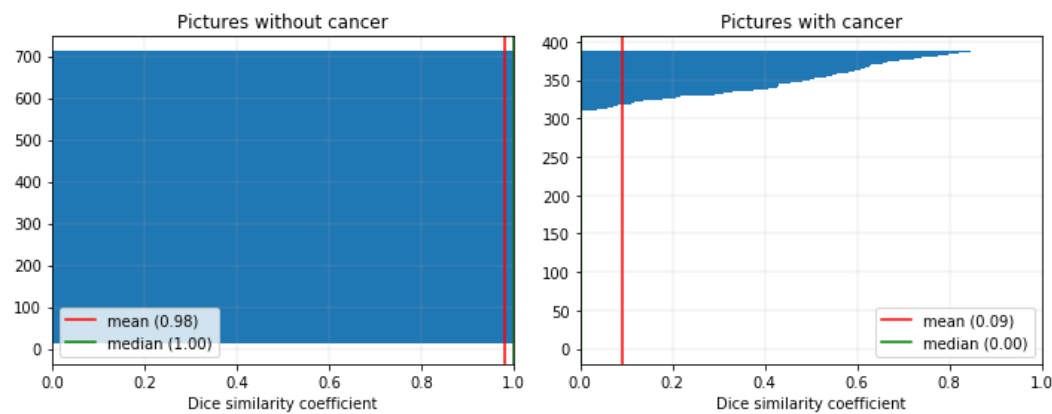


Rysunek 4.8

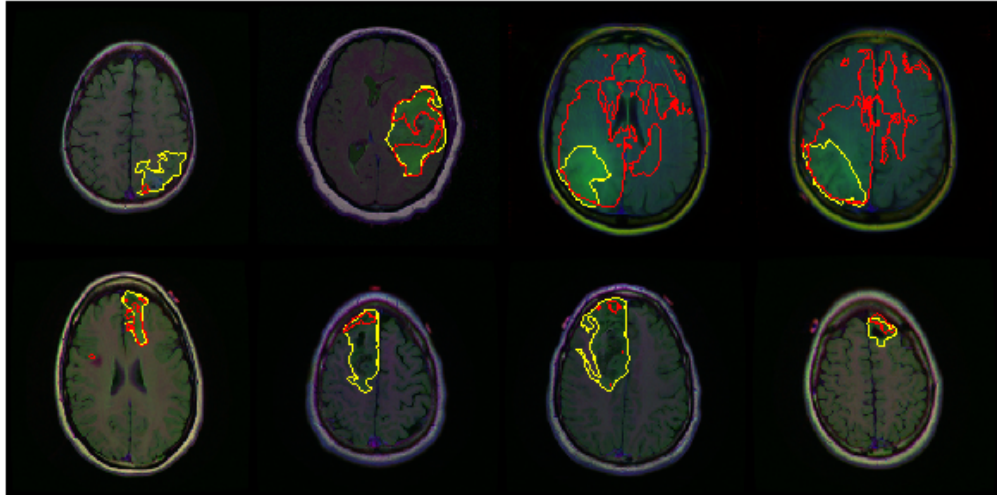
4.4. Back to basic models



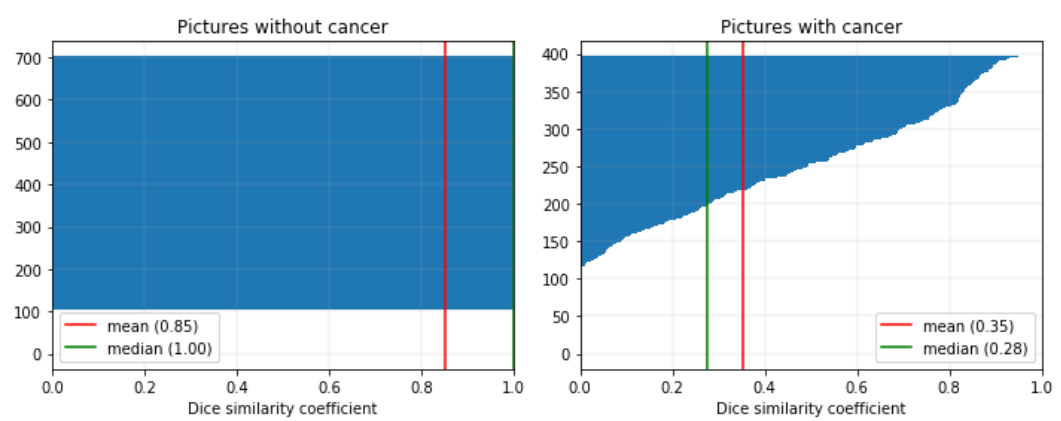
Rysunek 4.9



Rysunek 4.10



Rysunek 4.11



Rysunek 4.12



## Rozdział 5.

# Workflow



## Rozdział 6.

## Summary