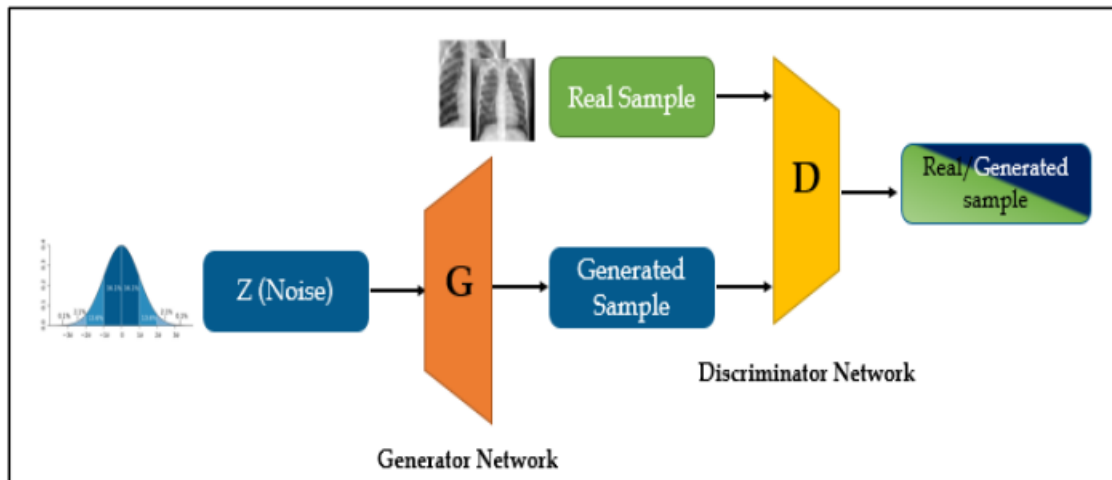


1 Augmentacja danych

1.1 Generative Adversarial Network

Generative Adversarial Network jest systemem dwóch sieci neuronowych - generatora i dyskriminatora, które wzajemnie się trenują poprzez generowanie przez generator danych zbliżonych do oryginalnego zbioru treningowego i ocenianiu jakości wygenerowanej próbki przez dyskriminator.



1.2 Literatura

W Detection of Coronavirus (COVID-19) Associated Pneumonia based on Generative Adversarial Networks and a Fine-Tuned Deep Transfer Learning Model using Chest X-ray Dataset autorzy dostarczają analizy, jak zastosowanie GANa przy różnych sieciach neuronowych przy zagadnieniach związanych z przetwarzaniem zdjęć rentgenowskich płuc (w tym także do diagnozy COVID-19) podniosło efektywność klasyfikacji.

1.3 Architektura do planowanej implementacji

Zaprezentowana w artykule architektura generatora GAN to:

- 5 transposed convolutional layers
- 4 ReLU layers
- 4 batch normalization layers
- 1 Tanh layer

Zaprezentowana w artykule architektura dyskriminatora GAN to:

- 5 convolutional layers
- 4 leaky ReLU layers

- 3 batch normalization layers

Powyższą architekturę planujemy zrealizować w Kerasie i dostosować do rozmiaru danych na których uczy się COVIDNet.

Użycie GANa przez autorów artykułu uniwersalnie zwiększyło efektywność modeli przede wszystkim przyczyniając się do zmniejszenia ryzyka przeuczenia.

Powyższe podejście zostało przetestowane na prostszych relatywnie niewielkich modelach, w przypadku COVIDNet możliwy jest brak jednoznacznie identyfikowalnej poprawy efektywności klasyfikacji.

2 Transfer learning (auxiliary task)

Naszym zadaniem jest stworzenie dodatkowego zadania dla naszej sieci neuronowej, które będzie na tyle adekwatne, że może pozwolić na szybsze oraz lepsze wytrenowanie sieci w kontekście jej głównego przeznaczenia. W celu najlepszego, realnego do zaimplementowania pomysłu dokonaliśmy przeglądu literatury.

2.1 Liebel, Lukas, and Marco Körner. "Auxiliary tasks in multi-task learning." arXiv preprint arXiv:1805.06334 (2018).

Pierwszym zbadanym artykułem był powyższy. Wskazywał on na benefits, jakie można uzyskać stosując multitask learning, ucząc w sposób jednoczesny kilka outputów z użyciem jednego inputu. Wykazywane jest eksperymentalne potwierdzenie poprawy jakości wyników oraz szybkości uczenia całej sieci. Artykuł wspomina o zastosowaniach dla konwolucyjnych sieci neuronowych i zadaniach typu image classification, sam jednak jest głównie skupiony na single-image depth estimation (SIDE) and semantic segmentation.

2.2 Malhotra, Aakarsh, et al. "Multi-Task Driven Explainable Diagnosis of COVID-19 using Chest X-ray Images." arXiv preprint arXiv:2008.03205 (2020).

Jest to kolejny przykład zastosowania multitaskingu, tym razem temat jest pokrewny do naszego. Zaproponowana przez autorów sieć nie tylko bada czy na zdjęciu są obecne oznaki, wskazujące na zarażenie COVID, a dodatkowo dokonuje segmentacji obrazu, co ma poprawić działanie sieci.

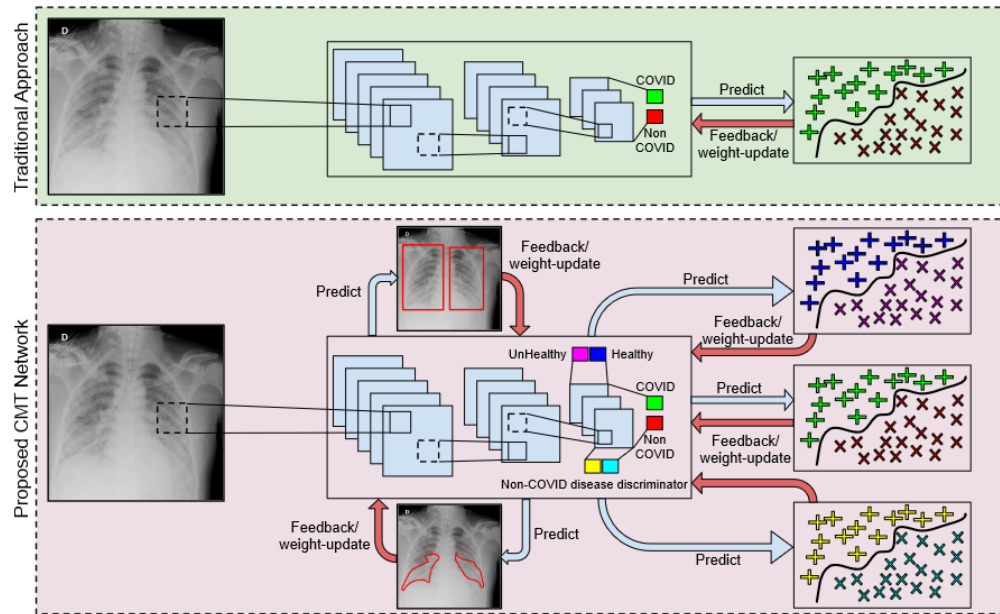


Fig. 2: The proposed CMTNet to perform multiple related tasks to improve the classification performance for COVID-19 disease diagnostics using frontal x-ray. The figure contrasts the multitask network with single task network.

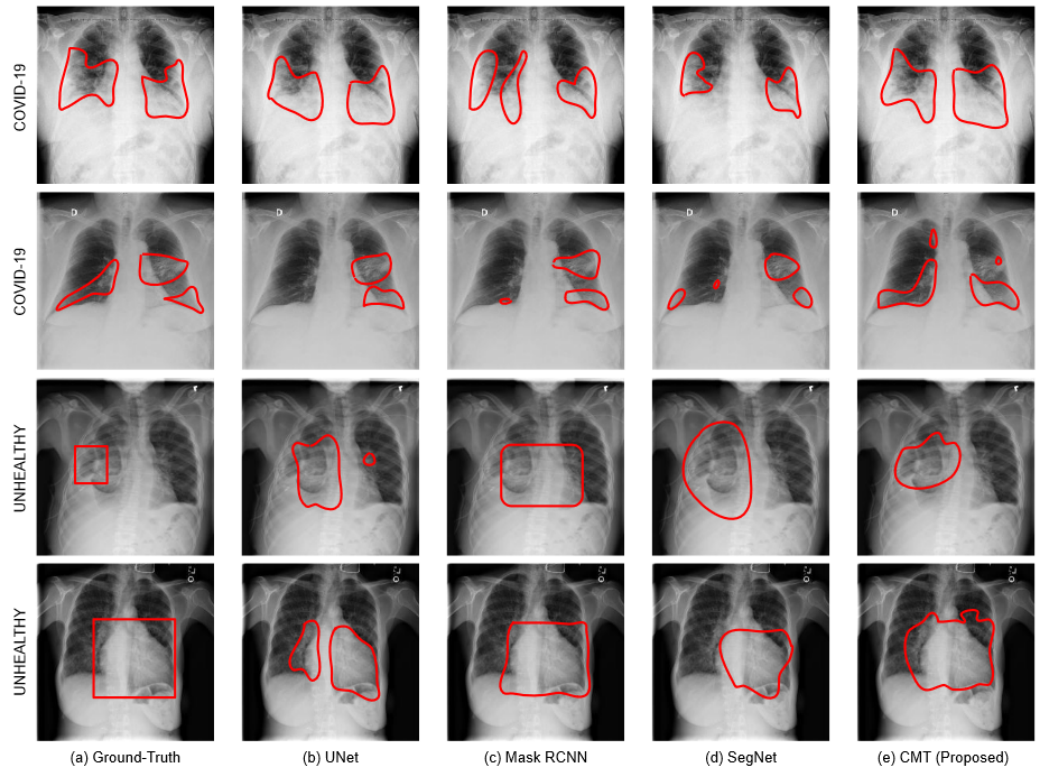


Fig. 6: Samples of semantic disease segmentation for existing algorithms and the proposed CMTNet. The x-ray images and corresponding abnormality localization for “Unhealthy” are derived from ChestXray-14 database [25].

2.3 Malhotra, Aakarsh, et al. "Multi-Task Driven Explainable Diagnosis of COVID-19 using Chest X-ray Images." arXiv preprint arXiv:2008.03205 (2020).

Również prezentuje ciekawe rozwiązanie. Autorom udało się poprawić działanie w przypadku obrazów X-ray o (0.96%-2.42%).

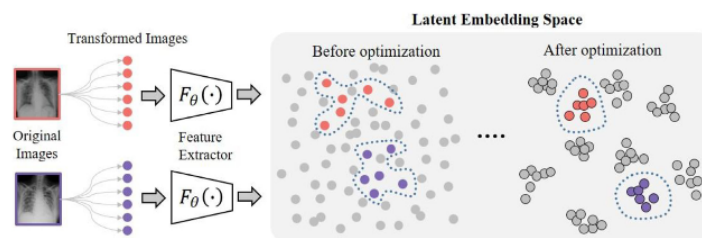


Fig. 3. Illustration of the effect of contrastive learning. We use different colors to distinguish different samples. Each image is transformed by a series of augmentation operations, and the feature extractor projects the transformed images to the latent embedding space. Contrastive learning is used to optimize the model for better spatial aggregating properties: representations belonging to a same image are concentrated together to formulate semantic clusters, whereas representations belonging to different images are scattered across the latent embedding space. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Obrazek powyżej pokazuje działanie contrastive learning. Po wyekstrahowaniu cech i transformacji do innej przestrzeni poprawiane są własności agregujące modelu.

2.4 Podsumowanie

Nie wszystkie z tych pomysłów wydają się jednak możliwe do zaimplementowania w naszej sieci neuronowej ze względu na trudności, jakie występują podczas pracy z nią. Ciężko oprzeć również pomysł o wykorzystanie metadanych, gdyż nie ma ich zbyt dużo. Pojawiły się zatem pomysły bazujące na wprowadzaniu zniekształceń do zdjęć, jednak tego też nie chcemy próbować na początku. Zatem pierwszym pomysłem, na auxiliary task jaki zamierzamy zaimplementować jest wykorzystanie zbioru <https://www.kaggle.com/nih-chest-xrays/data> i wytrenowanie naszego modelu do klasyfikacji zawartych tam chorób płuc.

3 Transfer learning (unsupervised pretraining)

3.1 MoCo-CXR: MoCo Pretraining Improves Representation and Transferability of Chest X-ray Models

Jest to dosyć świeży paper bo raptem z przed dwóch miesięcy. Autorzy pokazują przewagę użycia przetrenowanej części modelu w przypadku ImageNetu dla zdj X-ray klatki piersiowej. Wyniki pokazują, że model z użyciem tak zwanego Momentum Contrastive Learning pomaga uzyskać znacznie lepsze rezultaty gdy mamy dużą liczbę nie oznakowanych rekordów. Proces tego działania można zaobserwować na tym obrazku:

Takie podejście w pewnym sensie może pozwolić na użycie ImageNetu i dopasowanie takie modelu w przypadku gdy nie mamy dostępu do architektury sieci

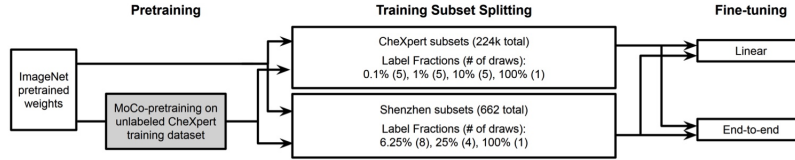


Figure 1: MoCo-CXR training pipeline. MoCo acts as self-supervised training agent. The model is subsequently tuned using chest X-ray images.

3.2 Klasyczne podejście unsupervised pre-training

W tym przypadku skorzystaliśmy z całkiem materiału video który objaśnia proces uczenia nie-nadzorowanego dla modelu. Jeśli chodzi o rzeczy związane z implementacją to są proponowane metody: Restricted Boltzmann machine lub np autoencoder.

Restricted Boltzmann machine

Autoencoder for lungs

Problem u nas w tym rozwiązaniu jest fakt że nie mamy dostępu do architektury sieci neuronowej więc będzie nam trudno wydzielić poszczególne warstwy i je uczyć.