

Sprawozdanie z laboratorium:  
Komunikacja Człowiek-Komputer

Wykrywanie zmian nowotworowych na obrazach wykonanych  
metodą rezonansu magnetycznego

9 grudnia 2020

Prowadzący: mgr inż. Agnieszka Mensfelt

	<b>Aleksandra Świerkowska</b>	141325
Autorzy:	<b>Michał Szczepaniak</b>	141317
	<b>Zuzanna Trafas</b>	141329

Zajęcia czwartkowe, 15:10-16:40

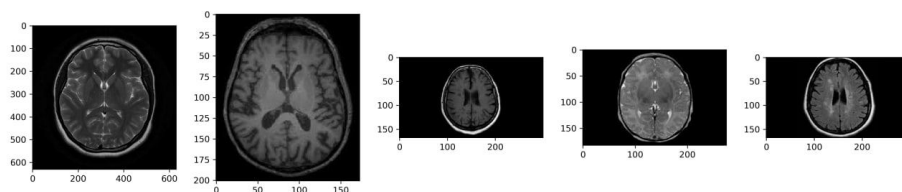
Oświadczamy, że niniejsze sprawozdanie zostało przygotowane wyłącznie przez powyższych autorów, a wszystkie elementy pochodzące z innych źródeł zostały odpowiednio zaznaczone i są cytowane w bibliografii.

# 1 Wstęp

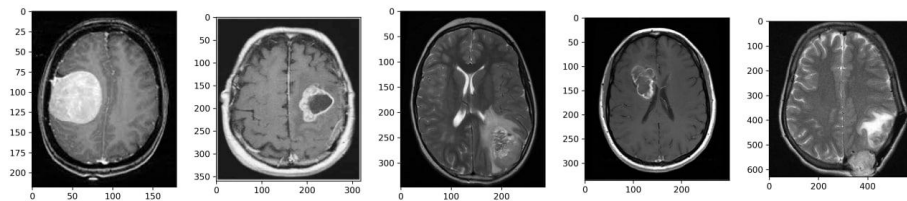
Celem projektu było napisanie i wytrenowanie konwolucyjnej sieci neuronowej w celu klasyfikacji zdjęć mózgu wykonanych metodą rezonansu magnetycznego na podstawie podejrzenia zmian nowotworowych. Michał Szczepaniak zajął się znalezieniem danych, przygotowaniem karty projektu oraz wycinaniem zdjęć, Aleksandra Świerkowska stworzyła raport oraz przygotowała dane, Zuzanna Trafas stworzyła sieć neuronową wraz z przetestowaniem jej działania za pomocą metody LIME.

## 2 Przygotowanie danych

Wykorzystane dane pochodzą ze zbioru “Brain MRI Images for Brain Tumor Detection”. Jest to zbiór dwustu pięćdziesięciu trzech zdjęć mózgu wykonanych za pomocą metody rezonansu magnetycznego<sup>1</sup>.



Rysunek 1: Przykłady zdjęć mózgów, w których nie znajdują się zmiany nowotworowe

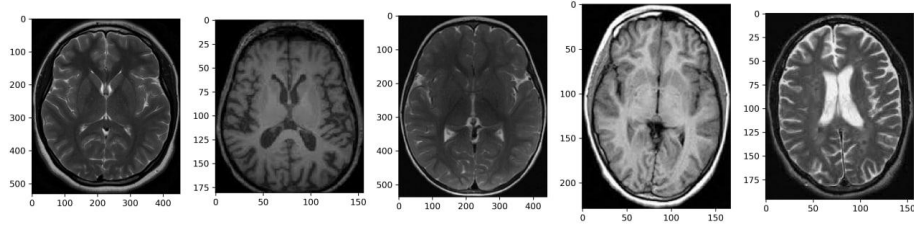


Rysunek 2: Przykłady zdjęć mózgów, w których znajdują się zmiany nowotworowe

Pierwszym etapem przygotowania danych było przycięcie zdjęć w celu pozbycia się zbędnego tła.

---

<sup>1</sup>Zgodnie z definicją Słownika języka polskiego PWN, rezonans magnetyczny to “zjawisko rezonansowego pochłaniania fal radiowych przez jądra atomowe znajdujące się w polu magnetycznym, wykorzystywane m.in. w medycynie do otrzymywania dokładnych obrazów tkanek i narządów wewnętrznych”.



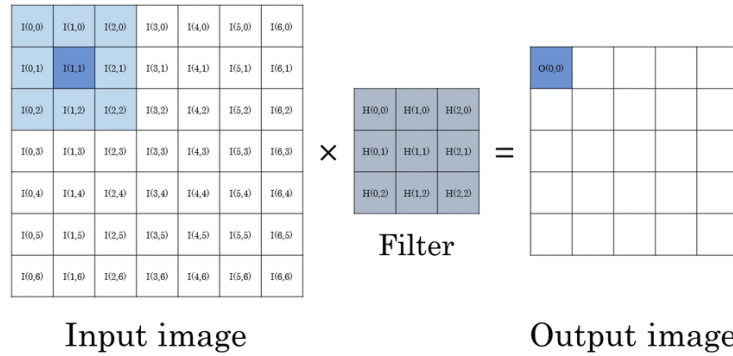
Rysunek 3: Przykłady zdjęć po przycięciu

Następnie dokonano podziału danych na zbiory treningowy, walidacyjny i testowy w proporcjach odpowiednio 75%, 20%, 5%. Wydzielone zdjęcia zapisano w odpowiednich folderach.

Ostatnim etapem przygotowania danych był proces augmentacji, niezbędny z uwagi na niewielki zbiór treningowy. Augmentacja polega na preparowaniu nowych zdjęć z już istniejących poprzez wprowadzanie drobnych zmian, takich jak rotacja, skalowanie, obrót w symetrii osiowej.

### 3 Sieć neuronowa

Konwulucyjne sieci neuronowe są stosowane głównie do przetwarzania obrazów. Nazwa tego rodzaju sieci pochodzi od operacji wykonywanej w jej warstwach, zwanej konwolucją (lub splotem) - operacji matematycznej, która polega na przesuwaniu filtra wzdłuż macierzy obrazu i wyznaczaniu oraz sumowaniu odpowiednich wartości:



Rysunek 4: Streaming Architecture for Large-Scale Quantized Neural Networks on an FPGA-Based Dataflow Platform - Scientific Figure on ResearchGate.

Przy tworzeniu naszej sieci wzorowaliśmy się na przykładowej architekturze

z oficjalnego blogu Keras - Building powerful image classification models using very little data<sup>2</sup>.

Podsumowanie warstw stworzonego przez nas modelu:

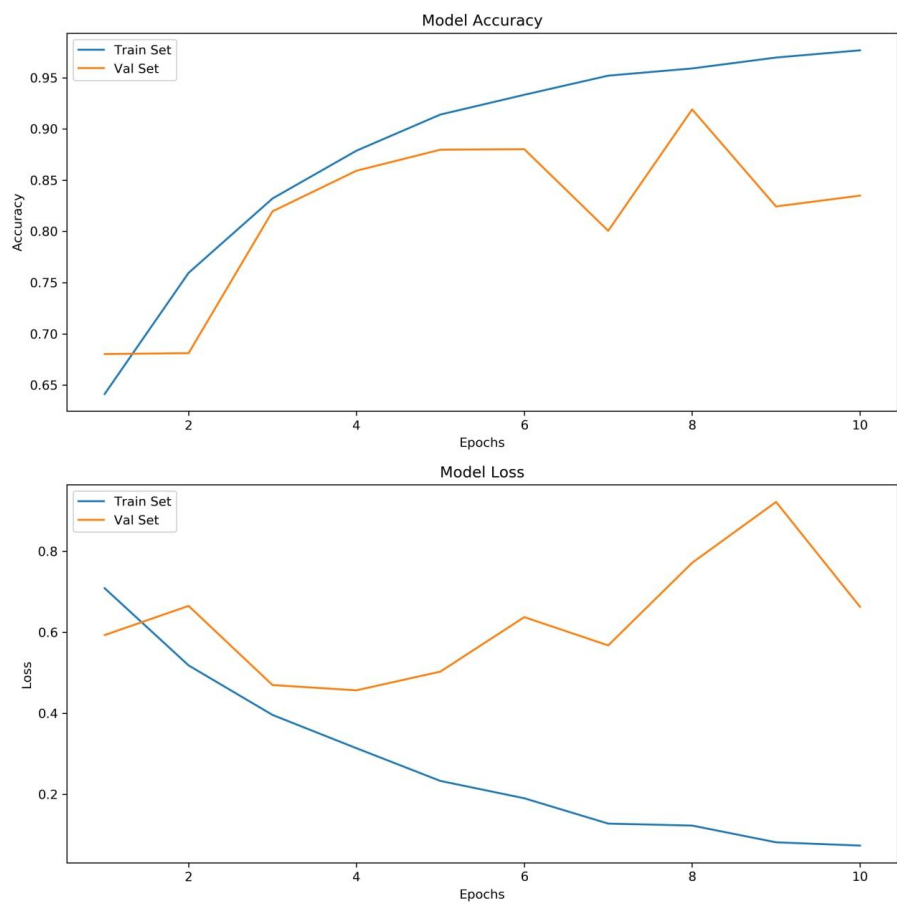
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_1 (MaxPooling2)	(None, 111, 111, 32)	0
conv2d_2 (Conv2D)	(None, 109, 109, 32)	9248
max_pooling2d_2 (MaxPooling2)	(None, 54, 54, 32)	0
conv2d_3 (Conv2D)	(None, 52, 52, 64)	18496
max_pooling2d_3 (MaxPooling2)	(None, 26, 26, 64)	0
flatten_1 (Flatten)	(None, 43264)	0
dense_1 (Dense)	(None, 64)	2768960
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65
Total params: 2,797,665		
Trainable params: 2,797,665		
Non-trainable params: 0		

Zgodnie z artykułem opublikowanym na blogu Keras, spróbowaliśmy trenować sieć przez 10 epok po 125 kroków w każdej (ok.15 min). Aby ocenić naszą sieć neuronową wykonywaliśmy wykresy dla accuracy oraz loss. Po wykonaniu wykresu dla pierwszej sieci, zauważyliśmy że sieć bardzo szybko zaczyna się przeczuć. [Rysunek 5]

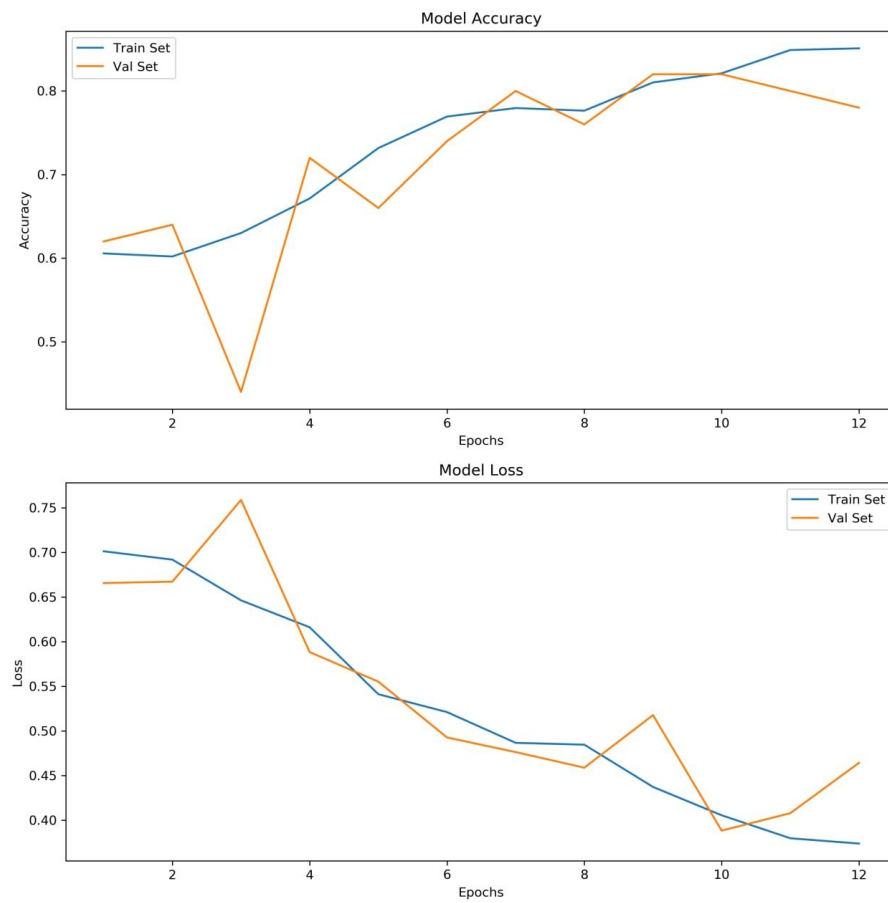
Próbowaliśmy zmniejszyć liczbę kroków oraz epok, jednak ciągle osiągaliliśmy niezadowalające wyniki. Wyniki udało się poprawić dzięki uproszczeniu danych - jako, że rezonans magnetyczny tworzy czarno-białe zdjęcia, zastąpienie reprezentacji RGB skalą szarości znacznie zmniejszyło liczbę parametrów modelu oraz poprawiło wyniki (12 epok po 31 kroków). [Rysunek 6]

Najlepsze wyniki otrzymaliśmy po zamianie optymalizatora “rmsprop” na optymalizator “adam”. [Rysunek 7]

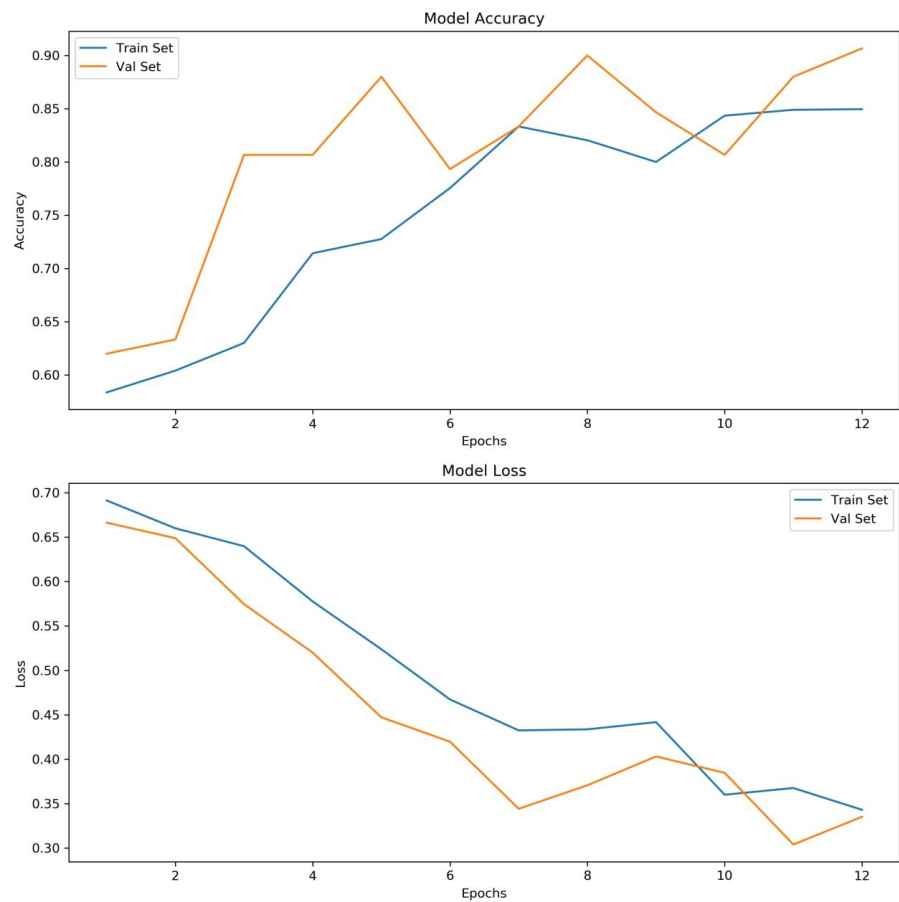
<sup>2</sup><https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>



Rysunek 5: Pierwotne wyniki



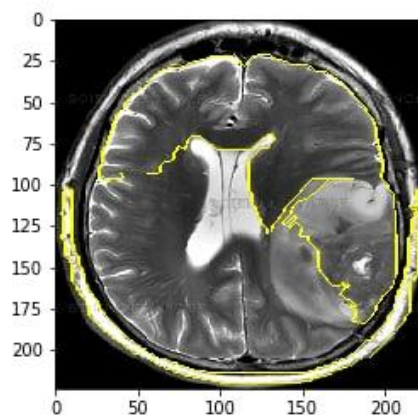
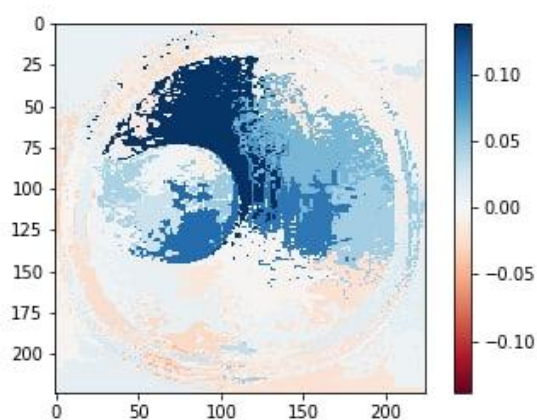
Rysunek 6: Wyniki po zastąpieniu reprezentacji RGB skalą szarości



Rysunek 7: Wyniki po zamianie optymalizatora "rmsprop" na optymalizator "adam"

## 4 Wyjaśnienie wyników klasyfikacji z wykorzystaniem LIME'a

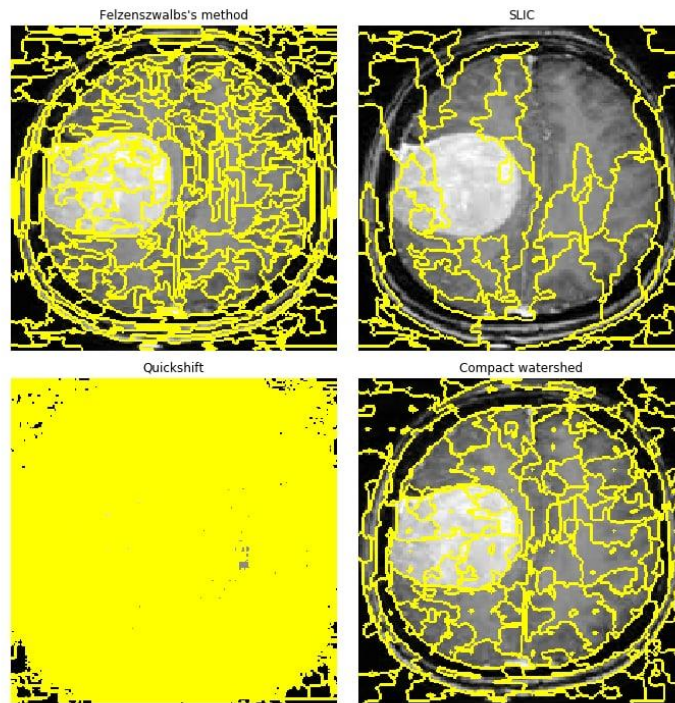
Postanowiliśmy wykorzystać metodę LIME do wytłumaczenia zachowania naszego modelu - metoda ta pozwala zaznaczyć, które obszary zdjęcia najbardziej wpływają na predykcję. Wzorowaliśmy się na przykładzie z oficjalnego repozytorium lime, rozwijanego przez jego twórcę - Mario Tulio Ribeiro<sup>3</sup>. Udało nam się uzyskać takie wytłumaczenia dla obrazu:



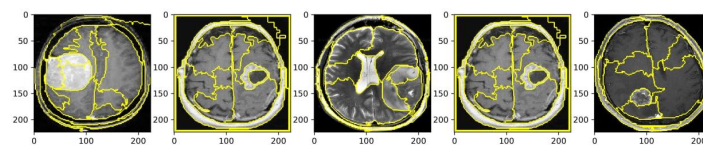
<sup>3</sup><https://marcotcr.github.io/lime/tutorials/Tutorial%20-%20images.html>



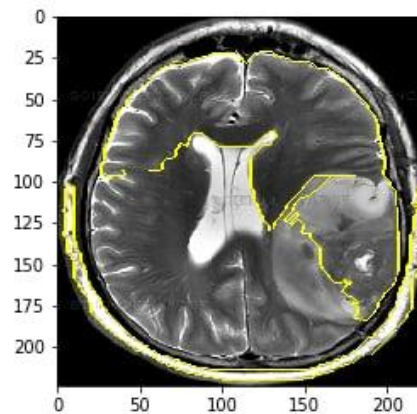
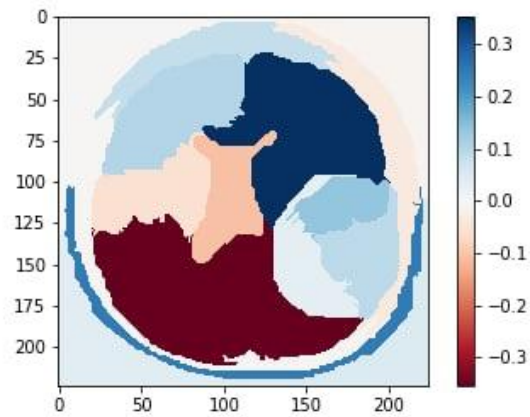
Założyliśmy, że może to wynikać z nieregularnego podziału na tzw. superpiksele - fragmenty obrazu. Postanowiliśmy stworzyć własną funkcję do segmentacji obrazu:



Wybraliśmy metodę Felzenszwalb'a, dobierając parametry do naszego zbioru danych. Po dobraniu parametrów udało nam się otrzymać taki podział:



Niestety nawet po zastosowaniu lepszego algorytmu do segmentacji, wyniki zwracane przez LIME'a wskazują głównie na mało istotne obszary:



Zakładamy iż może to wynikać z tego, że LIME analizuje jedynie niewielką ilość superpikeli naraz, zakrywając całą resztę. Guz sam w sobie nie zawiera żadnych punktów charakterystycznych, dużo łatwiej go zauważyć na tle mózgu. Z tego względu wyniki zwracane przez metodę LIME nie oznaczają obszaru z guzem jako najbardziej istotnego dla predykcji.

## 5 Wnioski

Udało się zbudować konwolucyjną sieć neuronową. Na danych testowych wartość *accuracy* wyniosła 84.3%. Nie udało się poprawnie wytłumaczyć działania sieci

metodą LIME, co wynikało ze specyfiki użytej metody i wybranych danych, nie z niepoprawnego działania samej sieci. Z tego samego powodu metoda SHAP, która potencjalnie mogłaby zastąpić metodę LIME, również nie przyniosłaby oczekiwanych rezultatów.

Największą trudnością okazała się mała ilość danych treningowych, przez co sieć bardzo szybko się przeuczała. Dzięki augmentacji danych możliwe było odpowiednie wytrenowanie sieci bez konieczności sięgania po *transfer learning*<sup>4</sup>.

## 6 Możliwe kierunki rozwoju

Stworzona sieć mogłaby uzyskać jeszcze lepsze wyniki po wytrenowaniu z wykorzystaniem większej ilości danych, a więc pierwszym krokiem w dalszym rozwoju powinno być powiększenie zbioru zdjęć.

Zwiększenie *accuracy* byłoby również możliwe dzięki wykorzystaniu wspomnianego wcześniej *transfer learningu*. Dużo łatwiej byłoby bowiem nauczyć już wytrenowaną sieć niż trenować od początku niewytrenowaną sieć mając do dyspozycji tak ograniczony zbiór danych.

Ponadto, projekt można by rozwinąć poprzez dodanie ręcznie stworzonych masek do danych oraz wytrenowanie osobnej sieci służącej do detekcji konkretnego obszaru zmian nowotworowych. W ten sposób sieć poza stwierdzaniem obecności zmian nowotworowych, byłaby również w stanie je wskazać. Umożliwiłoby to zastąpienie mało efektywnej metody LIME.

---

<sup>4</sup>Transfer Learning służy do ponownego wykorzystania wcześniej przygotowanego modelu do nowego problemu.

## Literatura

- [1] Lime documentation. URL: <https://lime-ml.readthedocs.io/en/latest/>.
- [2] Chaim Baskin. Streaming architecture for large-scale quantized neural networks on an fpga-based dataflow platform - scientific figure on researchgate, 2017. URL: [https://www.researchgate.net/figure/Image-convolution-with-an-input-image-of-size-7-7-and-a-filter-kernel-of-size-3-3\\_fig1\\_318849314](https://www.researchgate.net/figure/Image-convolution-with-an-input-image-of-size-7-7-and-a-filter-kernel-of-size-3-3_fig1_318849314).
- [3] Francois Chollet. Building powerful image classification models using very little data, 2016. URL: <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>.
- [4] Wikipedia. Convolutional neural network – wikipedia, the free encyclopedia. URL: [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network).
- [5] Victor Zhou. Cnns, part 1: An introduction to convolutional neural networks, 2019. URL: <https://victorzhou.com/blog/intro-to-cnns-part-1/>.