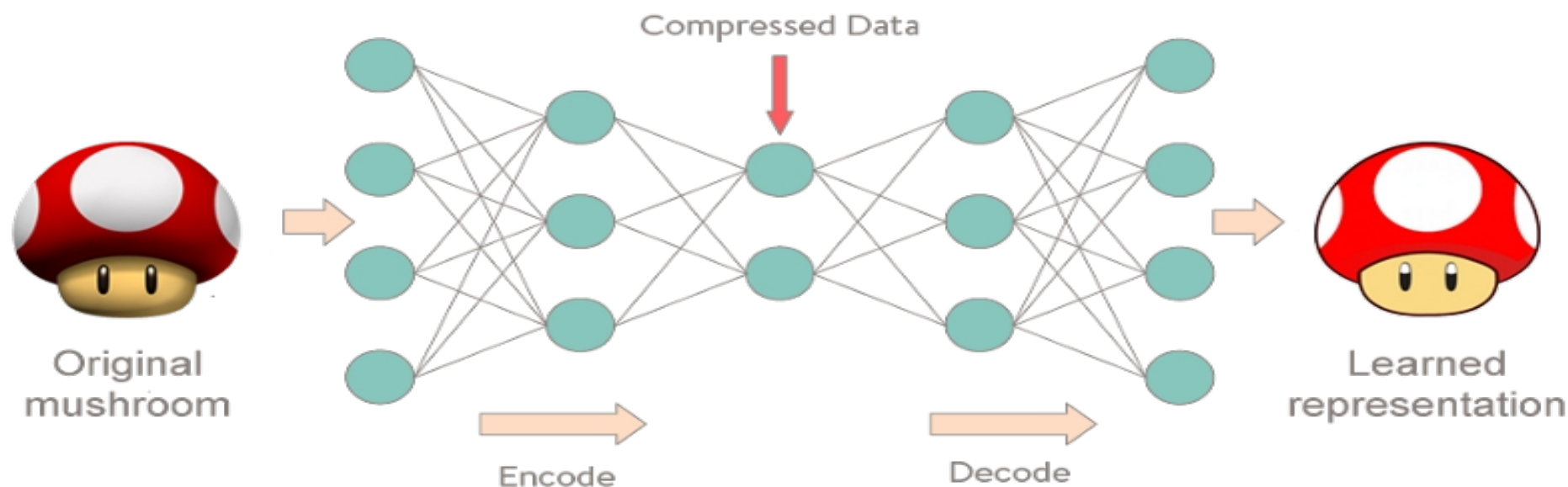


Klasyczny klasyfikator psów i kotów na podstawie cech generowanych przez NN

Plan

- 1) Zebrać dane: 25k otagowanych zdjęć.**
- 2) Wykorzystać już nauczoną sieć do wyciągnięcia istotnych cech z obrazków.**
- 3) Użycie PCA do redukcji wymiaru i potwierdzenia przydatności wyciągniętych cech.**
- 4) Wykorzystanie klasycznych algorytmów klasyfikacji jak Naive Bayes i SVM do klasyfikacji**
- 5) Walidacja skuteczności klasyfikatorów.**

Jak działają sieci neuronowe?

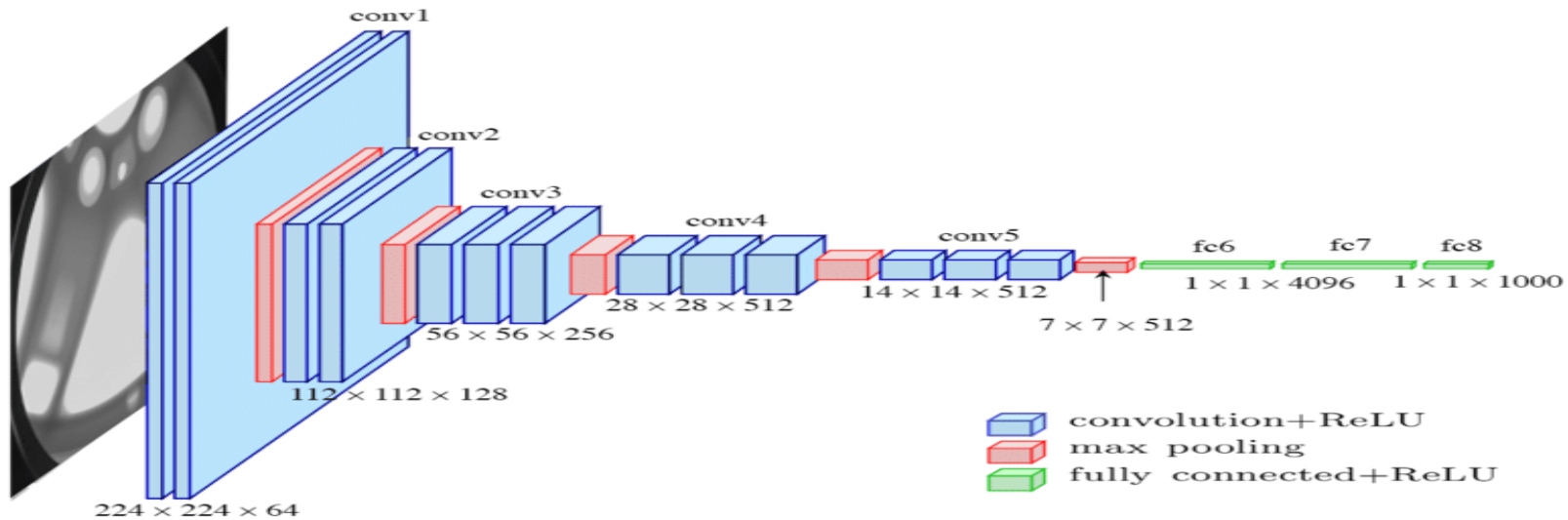


**Feature
extractor/
Encoder**

->

**Predictor/
Decoder**

VGG16



Params:

Feature extractor: 14M - Blue

Predictor: 120M - Green

Dlaczego może zadziałać?

Trenowany na:

ImageNet: 14M obrazków i 1000 klas

Trudny zróżnicowany dataset skuteczność ~70%, więc powinien się czegoś nauczyć.

Przykładowe klasy:

Cat:

281: 'tabby, tabby cat',

282: 'tiger cat',

283: 'Persian cat',

284: 'Siamese cat, Siamese',

285: 'Egyptian cat',

Dog:

245: 'French bulldog',

248: 'Eskimo dog, husky',

254: 'pug, pug-dog',

239: 'Bernese mountain dog',

934: 'hotdog, hot dog ;)',

Machine Learning Eldorado.

Bardzo dużo narzędzi open source, zbiorów danych i przetrenowanych modeli, platform:

Frameworks: Tensorflow, Keras

Datasets: Kagle, OPUS

Models: Transformer, VGG16, ResNet50, word2vec...

Platforms: colab, paperspace

RTX 2080 Ti

Price: 1200\$

Perf: 170 img/s

Train time: 24 days



Dataset labeled cats and dogs and preprocessing

1. Scale to model input dimensions

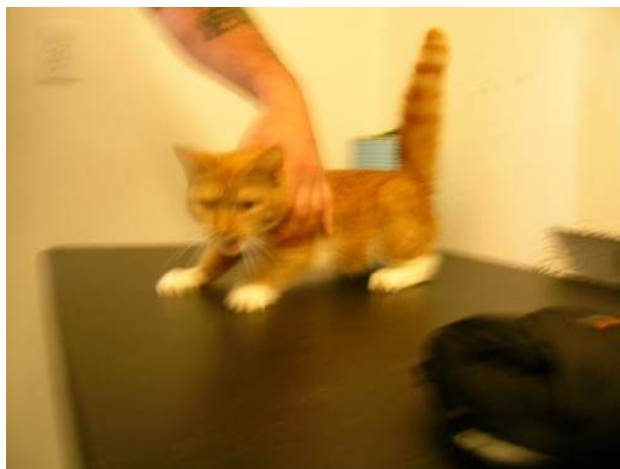
**2. Split channels
Red-Green-Blue**



Extracted features vectors: 25k dimensions

Pobrano zbiór
danych 25k zdjęć
kotów i psów

Ograniczyłem zbiór
treningowy do 1k, a
walidacyjny do 500
zdjęć.

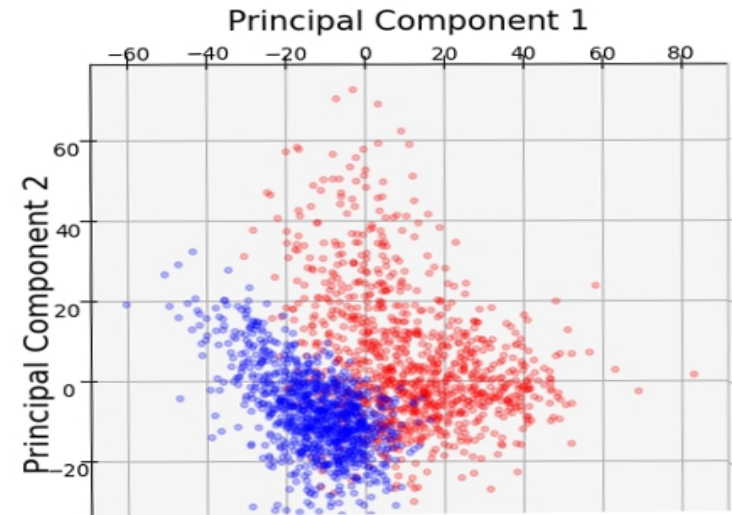
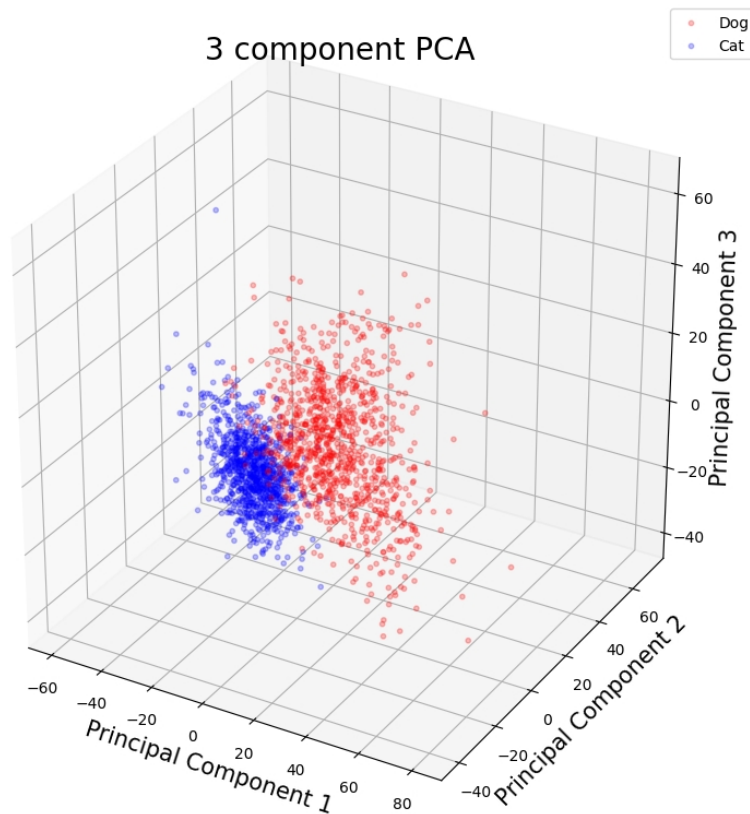


('cat',
[1.0,...,26.84,...,12.78,
...,2.32,3.89,...
.
.
.
7.24,...,28.95,...,
8.67,0.0])



('dog',
[1,...,10.55,...,32.26,
20.49,...,3.43,...
.
.
.
10.61,...,4.55,
5.88,0.0])

Predictor & PCA 3dim



PC1,	PC2,	PC3,	target
6.11,	-7.02,	5.78,	1
-6.70,	-12.84,	8.91,	0
26.28,	-3.84,	1.57,	1
-28.59,	5.90,	-7.85,	0
...			

Wyniki

Gaussian Naive Bayes:
Skuteczność 94.0%

SVM (rbf kernel):
Skuteczność 96.2%

Hipoteza:

- Za mało informacji, żeby poradzić sobie z powtarzającymi się wzorami, jak siatka/kraty.

Niepoprawna klasyfikacja dla obu metod:



Źródła:

medium.com/@franky07724_57962/visualize-image-clustering-fcd0dde89408

medium.com/@franky07724_57962/using-keras-pre-trained-models-for-feature-extraction-in-image-clustering-a142c6cdf5b1

Vgg16 paper: <https://arxiv.org/abs/1409.1556>

blog.skymind.ai/multi-tier-system-for-detecting-fraud-with-ai-and-deep-learning/

neurohive.io/en/popular-networks/vgg16/

lambdalabs.com/blog/2080-ti-deep-learning-benchmarks/