

Techniki uczenia maszynowego
stosowane w rozpoznawaniu wzorców
ze szczególnym uwzględnieniem metod
klasyfikacji znaków drogowych.

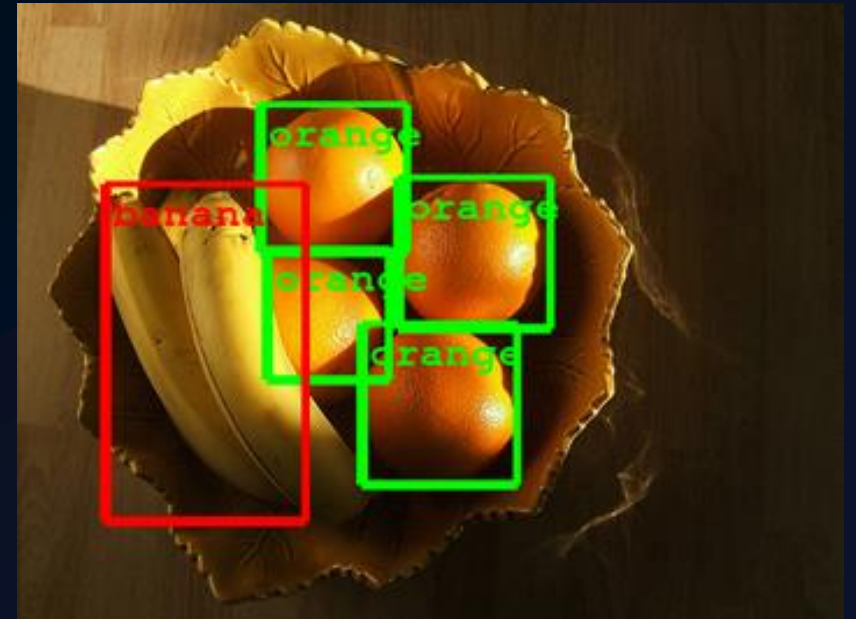
O czym będzie

- Rozpoznawanie wzorców (*ang. pattern recognition*)
- Obróbka obrazów i fake data – po co?
- Sieć neuronowa do wszystkiego?

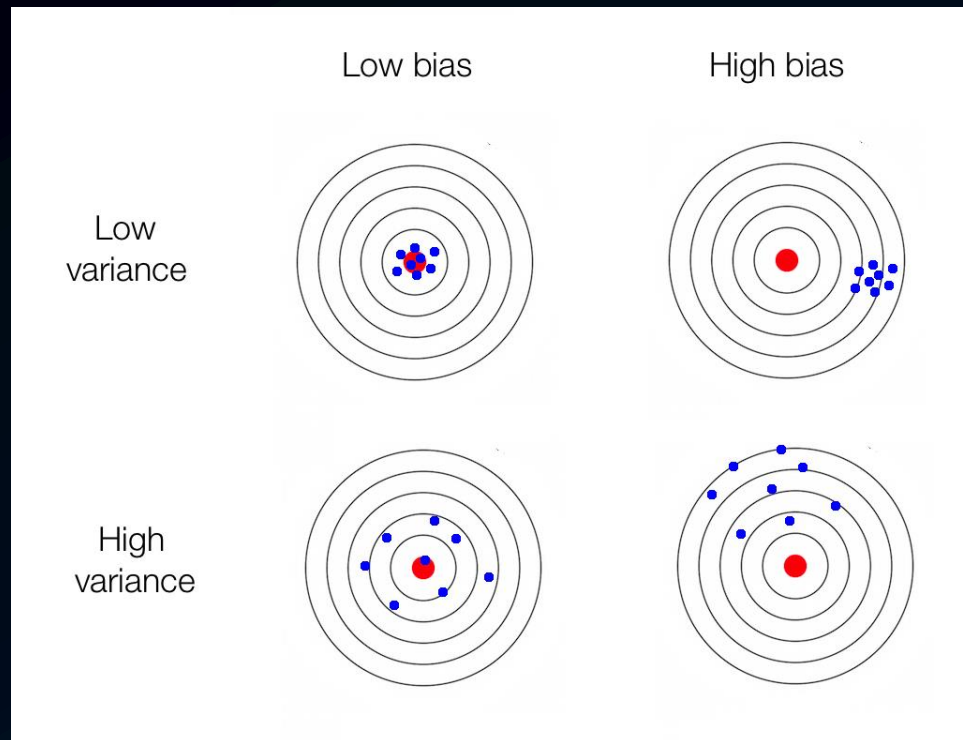
Rozpoznawanie a dopasowywanie wzorców

- Rozpoznawanie wzorców i dopasowywanie wzorców są czasami mylone jako takie same, gdy w rzeczywistości nie są.
- Rozpoznawanie wzorca poszukuje podobnego lub najbardziej prawdopodobnego wzorca w danych.
- Dopasowanie wzorca szuka dokładnie tego samego wzorca.

Dopasowywanie wzorców nie jest uważane za część uczenia maszynowego.



BIAS - odchylenie



W statystyce odchylenie estymatora jest różnicą pomiędzy oczekiwaną wartością estymatora a rzeczywistą wartością parametru, który jest szacowany.

PARADOKS DOKŁADNOŚCI

- paradoks dokładności – wybierając dane źródłowe chcemy, żeby były reprezentatywne, jeśli jednak rozkład częstości zmiennych wyjściowych jest niezrównoważony, czyli niektóre wartości występują znacznie rzadziej niż inne model jest bezużyteczny, ponieważ będzie odrzucał najmniej prawdopodobne scenariusze.

Jak działają algorytmy rozpoznawania obrazu?

- Odbywa się to w 3 krokach:
 - Pozyskanie obrazu,
 - Znalezienie matematycznego opisu obrazu,
 - Rozpoznanie i podjęcie decyzji.



Elementy składowe kompletnego systemu rozpoznawania obrazów:

- przetwarzanie niskiego poziomu - obejmuje pozyskanie obrazu, przetwarzanie wstępne, poprawę jakości obrazu, np. eliminacja zakłóceń, poprawa kontrastu, filtracja, normalizacja, fake data itd.,
- przetwarzanie średniego poziomu- dotyczy segmentacji obrazu oraz wydzielania i opisu cech obiektów obrazu, np. detekcja brzegów i konturów, przetwarzanie morfologiczne (dokładna analiza poszczególnych kształtów i odległości między nimi), itd.,
- przetwarzanie wysokiego poziomu-polega na klasyfikacji, rozpoznawaniu i interpretacji analizowanej sceny.

Wstępna obróbka obrazu – kod w Pythonie

```
import cv2
def grayAndEqualizeHist(img):
    """
    :param img: input RGB image
    :return: histogram equalized grayscale image
    """
    gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
    equ = cv2.equalizeHist(gray)
    # equ = equ - 128. / 128.
    return equ

x_train = np.array([grayAndEqualizeHist(img) for img in x_train])
x_test = np.array([grayAndEqualizeHist(img) for img in x_test])
```


Generowanie fake data

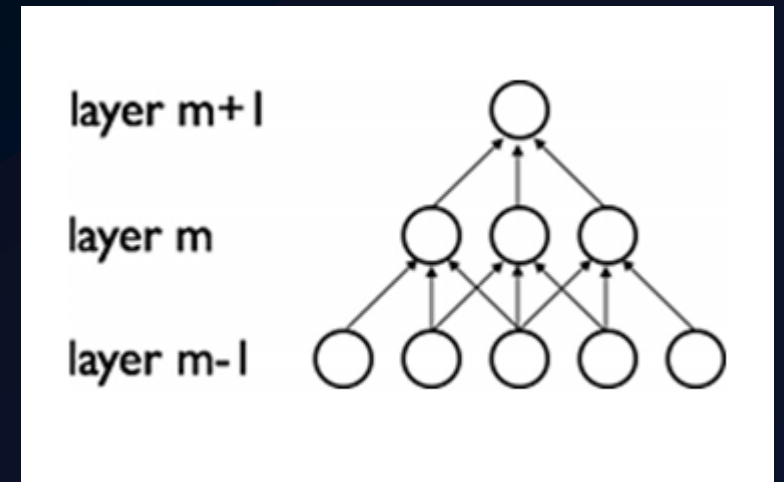
```
def applyRandSimilarityTran(image, n):  
    """  
        apply random similarity transformation to the image, and return  
        n transformed images  
    """  
    output_images = np.zeros((n, 32, 32))  
    for i in range(n):  
        angle = random.uniform(-15, 15) # rotation  
        rows, cols = image.shape[0:2]  
        image_center = (rows/2.0, cols/2.0)  
        rot_mat = cv2.getRotationMatrix2D(image_center, angle, 1)  
        M_rot = np.vstack([rot_mat, [0, 0, 1]])  
        tx = random.uniform(-2, 2) # translation along x axis  
        ty = random.uniform(-2, 2) # translation along y axis  
        M_tran = np.float32([[1, 0, tx], [0, 1, ty], [0, 0, 1]])  
        M = np.matrix(M_tran) * np.matrix(M_rot)  
        M = np.float32(M[:2][:]) # similarity transform  
        tmp = cv2.warpAffine(image, M, (cols, rows))  
        output_images[i][:][:] = tmp  
  
    return output_images
```

Sieci neuronowe

DEDYKOWANE DO ROZPOZNAWANIA
WZORCÓW

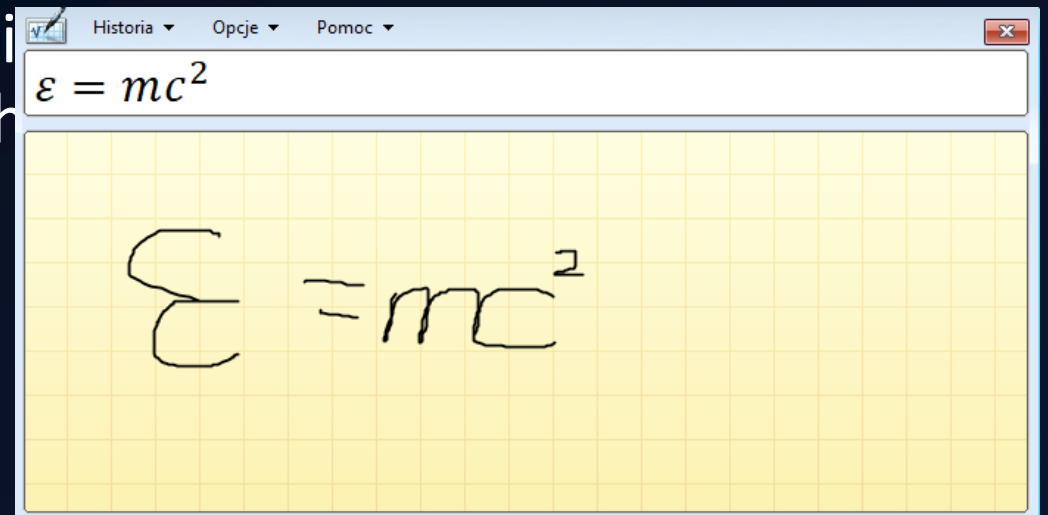
ConvNet (Convolutional Neural Networks)

- Konwolucyjna sieć neuronowa (CNN lub ConvNet) to specjalny rodzaj wielowarstwowych sieci neuronowych, zaprojektowanych do rozpoznawania wzorców wizualnych bezpośrednio z obrazów pikseli przy minimalnym wstępnym przetwarzaniu.
- Dane wejściowe jednostek ukrytych w warstwie **m** pochodzą z podzbioru jednostek w warstwie **m-1**

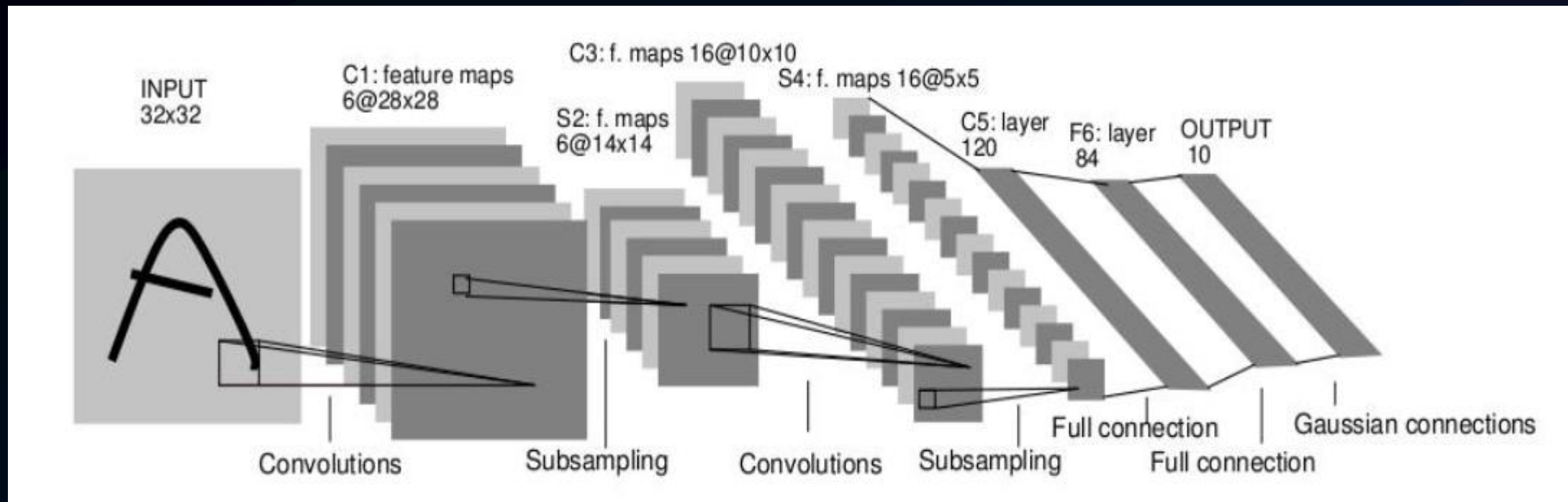


LeNet-5

- Pionierska, 7-poziomowa sieć splotowa z 1998 r., która klasyfikuje znaki.
- Została zastosowana przez kilka banków, aby rozpoznać odręczne liczby na czekach poddanych cyfryzacji w obrazach wejściowych w skali szarości 32x32 piksele. Zdolność przetwarzania obrazów o wyższej rozdzielczości wymaga więcej warstw, ale jest idealna do prostych rozpoznawanie odręcznego pisma.



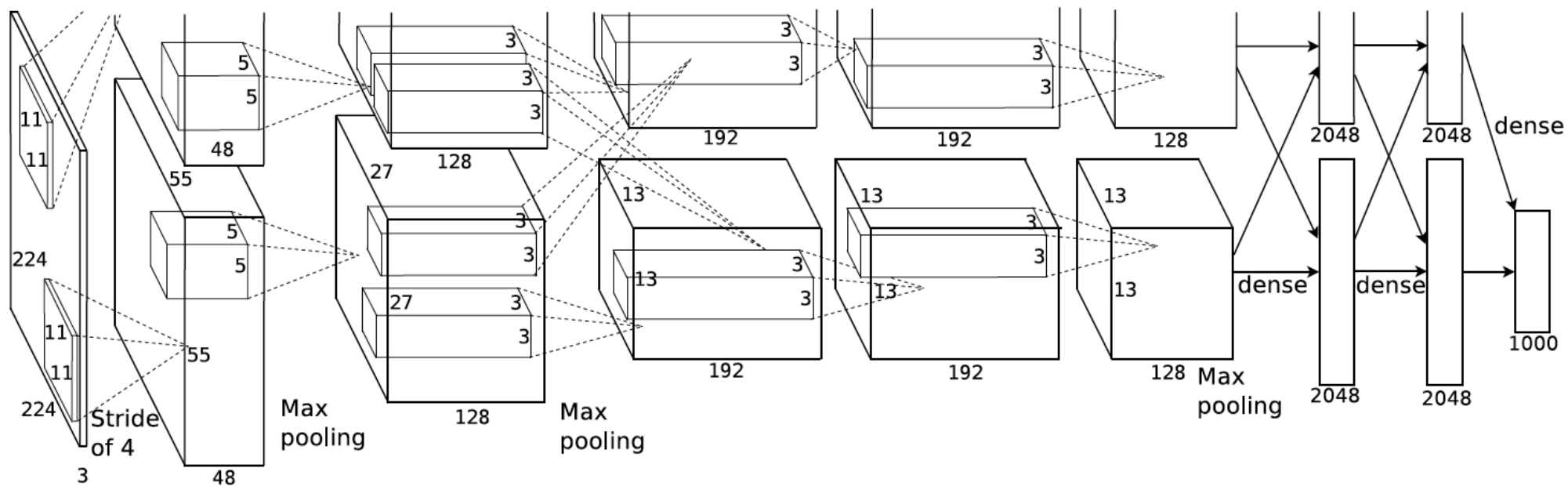
Model warstw sieci LeNet-5



AlexNet

- AlexNet zwyciężył w 2012 roku w konkursie ImageNet LSVRC-2012 z marginesem błędu 15,3% (dla człowieka to 5,1%)
- Zawiera 5 warstw splotowych i 3 warstwy całkowicie połączone, obraz wejściowy może wynosić 227x227
- Wykorzystuje się ją w medycynie np. do klasyfikowania czerniaków (złośliwe czy łagodne) lub rozpoznawania guzów w klatce piersiowej.

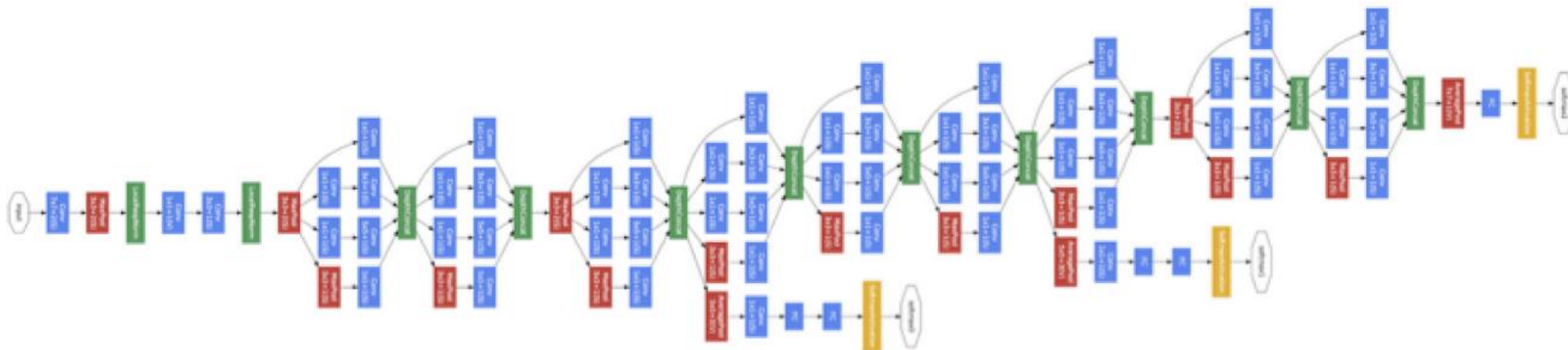
Model warstw sieci AlexNet



GoogleNet/Inception

- Został zwycięzcą ILSVRC-2014 osiągając poziom błędu rzędu 6,67%, który jest bardzo zbliżony do osiągnięć na poziomie ludzkim, które wynoszą 5,1%
- Odkryto, że równoważąc liczbę filtrów na warstwę i głębokość sieci można osiągnąć optymalną wydajność sieci

Model warstw sieci Inception



Convolution
Pooling
Softmax
Other

KONKLUZJA

HONDA SENSING®

- System wspomagania jazdy:
 - System rozpoznawania znaków drogowych
 - Asystent trzymania się linii,
 - Adaptacyjny tempomat po wykrytych pojazdach
 - System regulujący hamowanie i kierowanie, jeśli wykryje, że przekraczasz linie bez zasygnalizowania tego
 - Stosuje ciśnienie hamowania, gdy zostanie stwierdzona nieunikniona kolizja

Źródła

- https://en.wikipedia.org/wiki/Pattern_recognition
- <http://pe.org.pl/articles/2016/1/10.pdf>
- <https://ksopyla.com/python/tensorflow/klasyfikacja-cyfr-mnist-tensorflow/>
- <https://ksopyla.com/machine-learning/siec-konwolucyjna-rozpoznawanie-cyfr-z-obrazow/>
- <https://automobiles.honda.com/sensing>
- <https://www.pyimagesearch.com/2016/08/01/lenet-convolutional-neural-network-in-python/>