

6. Arhitecturi de rețele neuronale convoluționale

Arhitecturi clasice:

- LeNet-5
- AlexNet
- VGG-16

Arhitecturi moderne:

- Inception
- ResNet
- DenseNet

LeNet-5

Modelul LeNet-5 (LeCun et al., 1998) a fost dezvoltat pentru identificarea cifrelor scrise de mână pentru recunoașterea codurilor poștale.

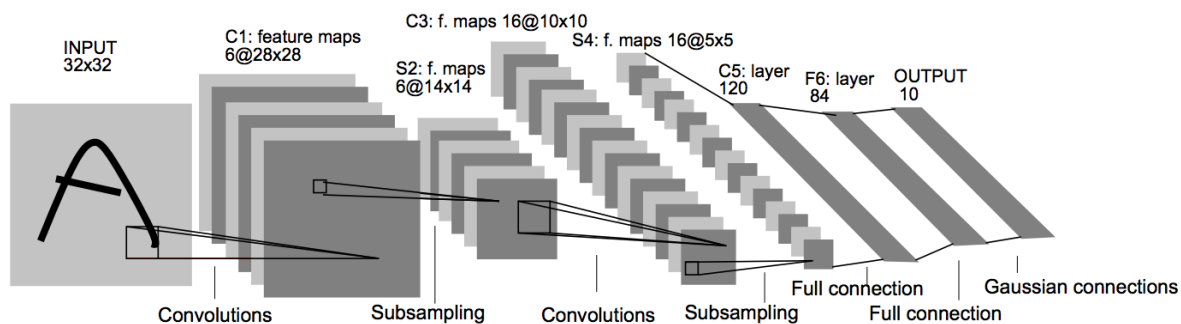


Fig. 6.1 Arhitectura modelului LeNet-5 (LeCun et al., 1998)

AlexNet

Modelul AlexNet (Krizhevsky et al., 2012) a ocupat primul loc în competiția ImageNet din 2012.

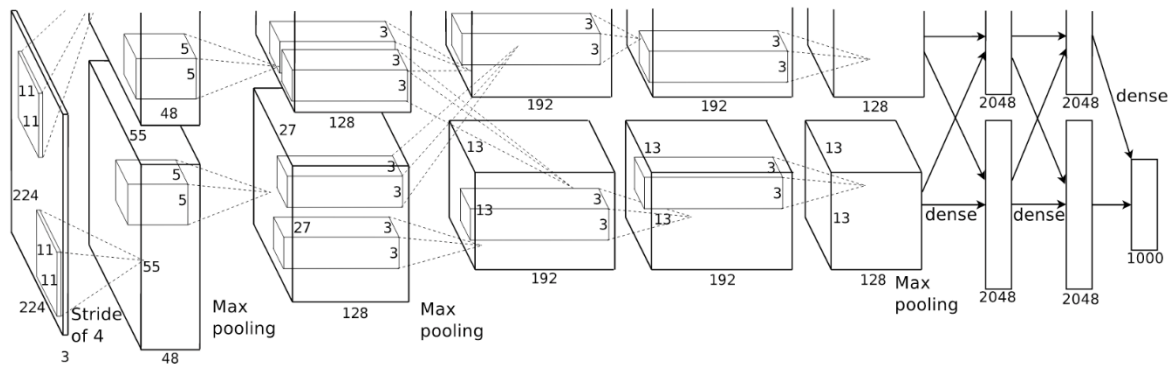


Fig. 6.2 Arhitectura modelului AlexNet (Krizhevsky et al., 2012)

VGG-16

(Simonyan & Zisserman, 2014)

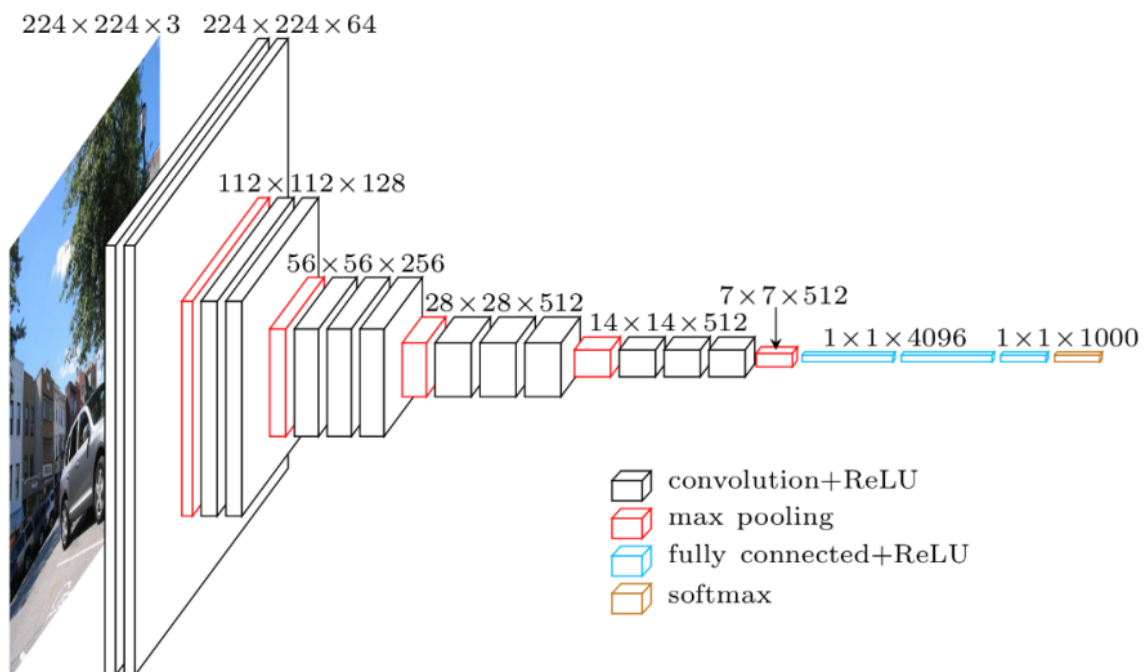


Fig. 6.3 Arhitectura modelului VGG-16 (Jordan, 2018)

Inception (GoogLeNet)

Rețeaua Inception (Szegedy et al., 2015) a obținut primul loc în competiția ImageNet 2014.

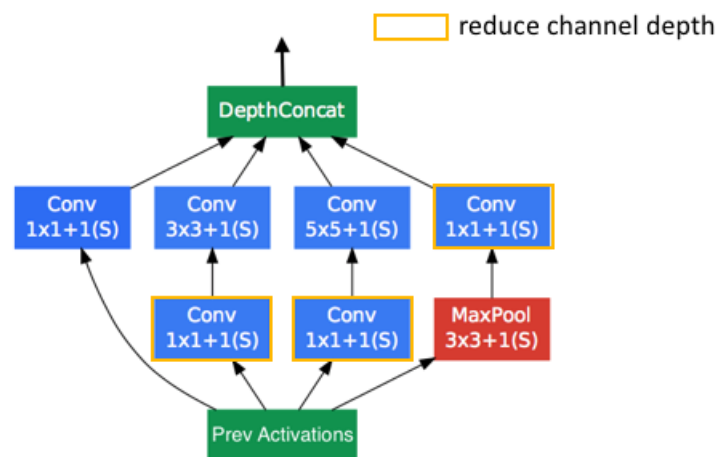


Fig. 6.4 Un modul Inception (Jordan, 2018)

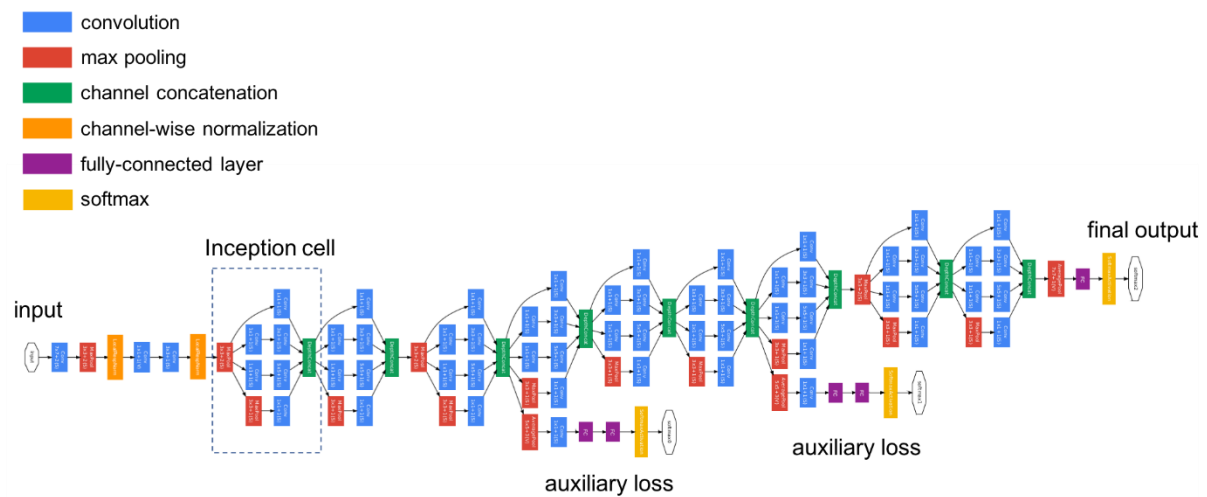


Fig. 6.5 Rețeaua GoogLeNet (Szegedy et al., 2015)

ResNet

He et al. (2016) au câștigat competiția ILSVRC 2015 folosind o rețea reziduală, ResNet.

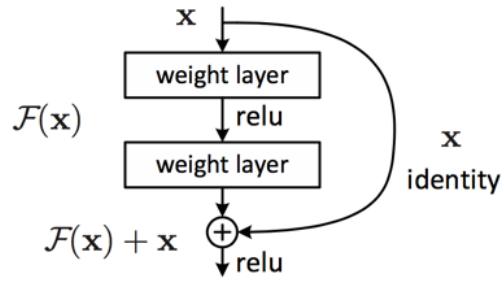


Fig. 6.6 Un bloc rezidual (He et al., 2016)

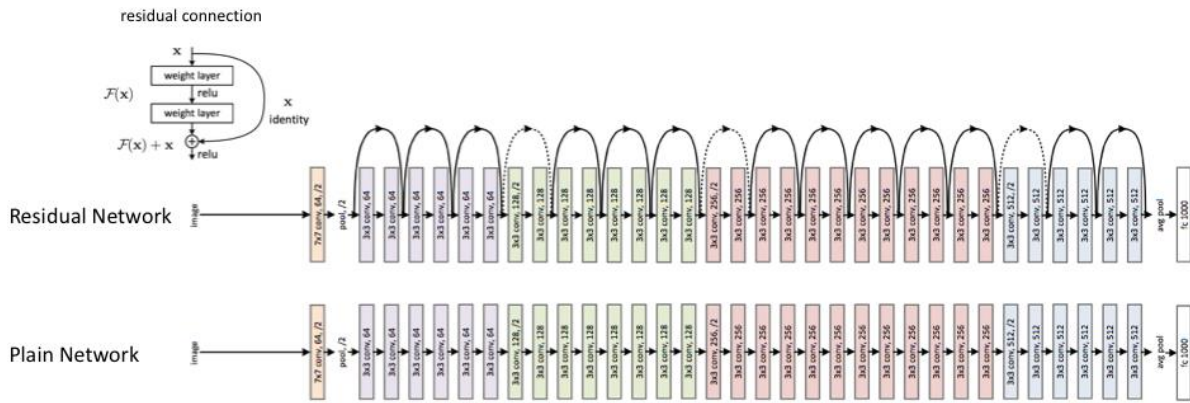


Fig. 6.7 Arhitectura modelului ResNet 34 (He et al., 2016)

DenseNet

În cazul rețelelor convoluționale dens conectate (Huang et al., 2017), harta caracteristicilor fiecărui strat este concatenată la intrarea fiecărui strat ulterior din cadrul unui bloc dens.

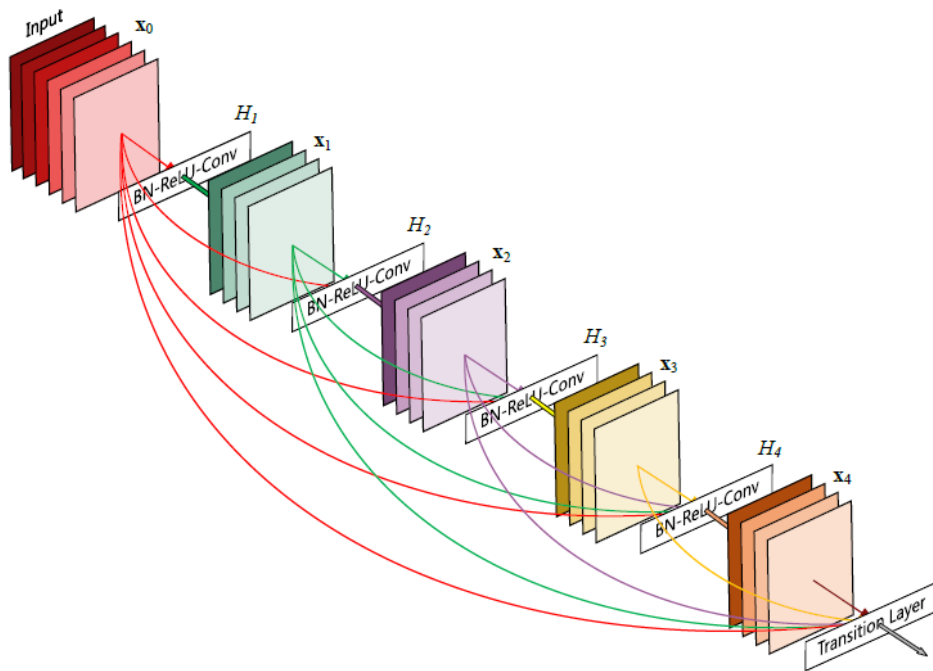


Fig. 6.8 Un bloc dens cu 5 straturi (Huang et al., 2017)

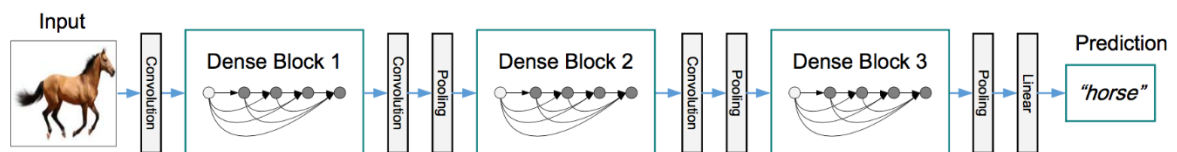


Fig. 6.9 O rețea DenseNet profundă cu trei blocuri dense (Huang et al., 2017)

Utilizarea modelelor preantrenate din Keras

Exemplu de utilizare a modelului ResNet-50, preantrenat pe ImageNet:

```
from sklearn.datasets import load_sample_images

images = load_sample_images()["images"]
```

Image: china.jpg



Image: flower.jpg



```
Image #0 shape: (427, 640, 3)
Image #1 shape: (427, 640, 3)
```

```
import tensorflow as tf

model = tf.keras.applications.ResNet50(weights='imagenet')

images_resized = tf.keras.layers.Resizing(height=224, width=224,
                                           crop_to_aspect_ratio=True)(images)

inputs = tf.keras.applications.resnet50.preprocess_input(images_resized)

preds = model.predict(inputs)
print(preds.shape)
```

```
(2, 1000)
```

```
top_K = tf.keras.applications.resnet50.decode_predictions(preds, top=3)
for image_index in range(len(images)):
    print(f"Image #{image_index}")
    for class_id, name, y_proba in top_K[image_index]:
        print(f" {class_id} - {name:12s} {y_proba:.2%}")
```

```
Image #0
n03877845 - palace          54.69%
n03781244 - monastery      24.71%
n02825657 - bell_cote      18.55%

Image #1
n04522168 - vase           32.67%
n11939491 - daisy          17.82%
n03530642 - honeycomb      12.04%
```

Învățarea prin transfer

