HỌC SÂU

PGS. TS. Nguyễn Hoài Nam

Trưởng NCM, Trưởng Lab (MASC)
Khoa Tự động hóa - Trường Điện Điện tử
Đại học Bách Khoa Hà Nội

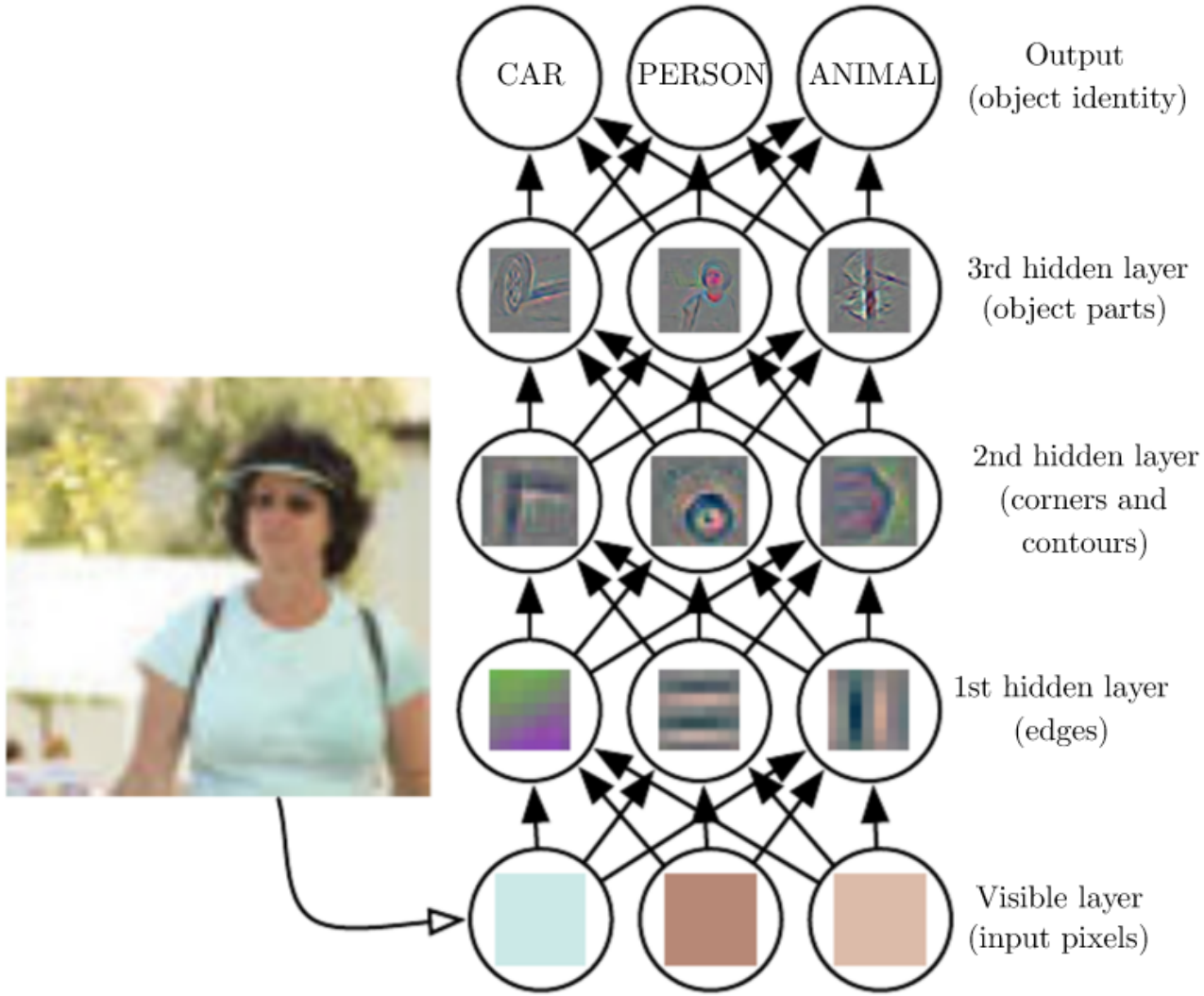
Email: nam.nguyenhoai@hust.edu.vn
Website: https://sites.google.com/view/masc-lab

Ngày 22 tháng 10 năm 2025

Chương 3 Mạng nơ-ron tích chập và ứng dụng

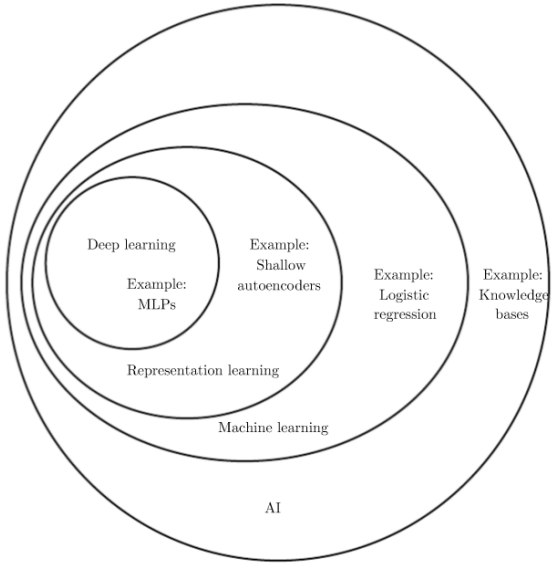
- 3.1. Khái niệm về học sâu
- 3.2. Tích chập và trích chọn đặc trưng
- 3.3. Mạng LeNet 5
- 3.4 Một số vấn đề CNN
- 3.5. Một số mô hình hiện đại dựa trên CNN
- 3.6. Ứng dụng phân loại và nhận dạng
- 3.7. Một số kỹ thuật trong nhận dạng

3.1. Khái niệm về học sâu



Hình 3.1: Mô hình mạng sâu

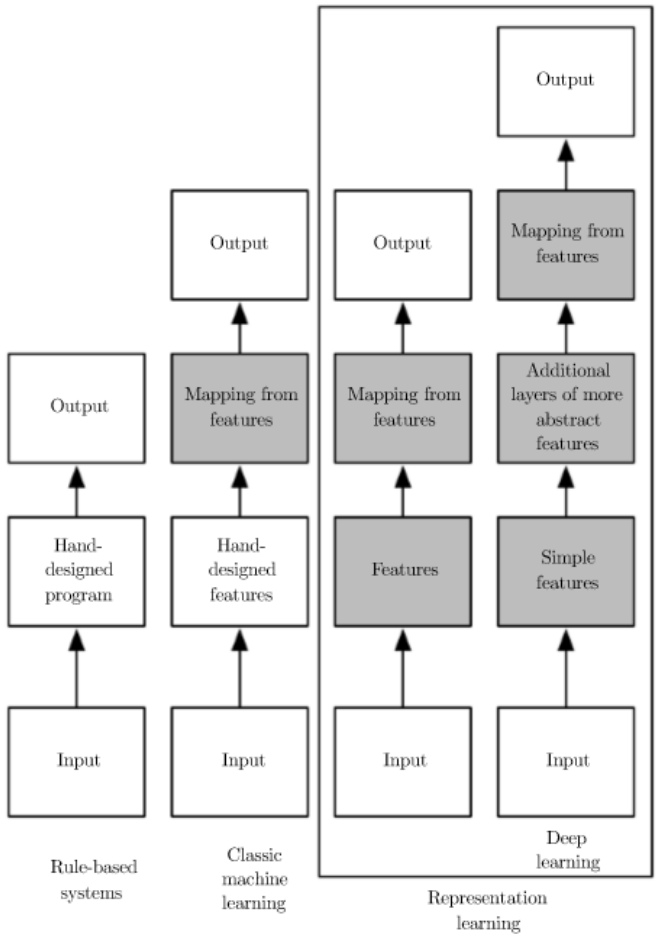
3.1. Khái niệm về học sâu



Hình 3.2: Học sâu và AI

- MLPs: Mạng nhiều lớp
- Autoencoder (it lớp): Mạng truyền thẳng học không có giám sát, dùng để tái tạo dữ liệu đầu vào, giảm chiều dữ liệu, khử nhiễu từ ảnh, phát hiện bất thường.
- Logistic Regression: Phân loại, xấp xỉ.
- Knowledge bases: Các cơ sở tri thức.

3.1. Khái niệm về học sâu



Hình 3.3: Mối liên hệ giữa các khối trong hệ thống AI

3.2. Tích chập và trích chọn đặc trưng

Tích chập (continuous-time):

$$\begin{aligned} s(t) &= \int_{\alpha}^{\beta} x(\tau) w(t-\tau) d\tau \\ &= (x * w)(t) \end{aligned} \tag{3.1}$$

Tích chập (discrete-time):

$$\begin{aligned} s(t) &= \sum_{\tau=\alpha}^{\beta} x(\tau) w(t-\tau) d\tau \\ &= (x * w)(t) \end{aligned} \tag{3.2}$$

x: đầu vào (input), w bộ lọc (kernel), và s: đặc trưng (feature map). Convolution có tính chất giao hoán (commutative).

3.2. Tích chập và trích chọn đặc trưng

Ảnh hai chiều:

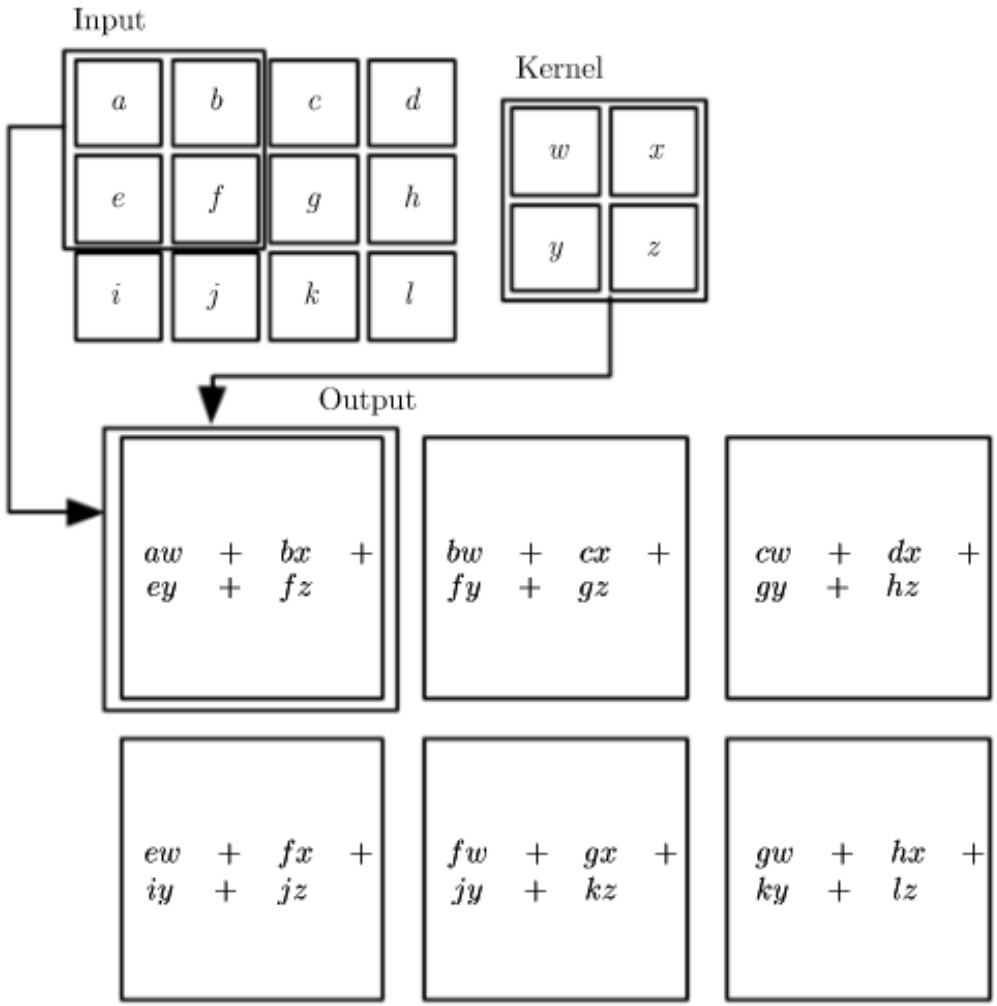
$$\begin{aligned} S(i,j) =& I * K(i,j) \\ =& \sum_m \sum_n I(m,n) K(i-m, j-n) \\ =& K * I(i,j) \end{aligned} \tag{3.3}$$

3.2. Tích chập và trích chọn đặc trưng

Trong mạng nơ-ron, tích chập (cross-correlation):

$$\begin{aligned} S(i,j) &= I * K(i,j) \\ &= \sum_m \sum_n I(i+m, j+n) K(m,n) \end{aligned} \tag{3.4}$$

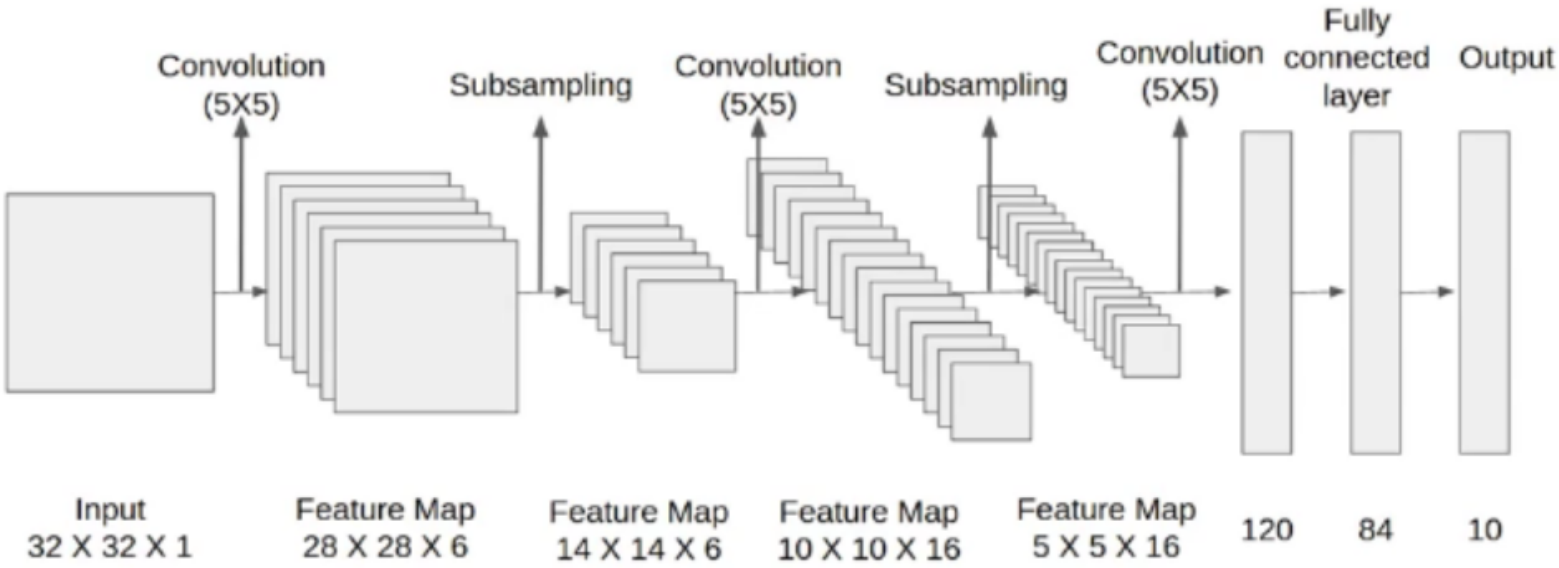
3.2. Tích chập và trích chọn đặc trưng



Hình 3.4: Ví dụ về tích chập 2D

3.3. Mạng LeNet 5

- Ra đời 1998[1]



Hình 3.5: Cấu trúc mô hình mạng Lenet 5

[1] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

3.3.1 Các lớp của mạng LeNet 5

- Đầu vào: 32×32 điểm ảnh
- Lớp tích chập $C1$: 6 ảnh đặc trưng (feature map). Mỗi đầu ra của lớp liên kết với một vùng 5×5 ở đầu vào. Kích thước ảnh đặc trưng 28×28. Lớp gồm 156 tham số và 122304 kết nối.
- Lớp nhóm S2: 6 ảnh đặc trưng 14×14. Mỗi đầu ra của ảnh liên kết với vùng 2×2 ở đầu vào. Tổng của 4 đầu vào được nhân với 1 tham số và cộng với bias. Kết quả được đưa vào hàm *sigmoid*. Lớp có 12 tham số và 5880 kết nối.

3.3.1 Các lớp của mạng LeNet 5

- Lớp $C3$: 16 ảnh đặc trưng. Mỗi đầu ra của ảnh đặc trưng được nối với vùng 5×5 ở những vị trí giống nhau trong tập con các ảnh đặc trưng $S2$. $C3$ có 1516 tham số và 156000 kết nối.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

Hình 3.6: Mối quan hệ giữa $S2$ và $C3$.

3.3.1 Các lớp của mạng LeNet 5

- Lớp $S4$ có 16 ảnh đặc trưng 5×5, 32 tham số và 2000 kết nối.
- Lớp $C5$ có 120 ảnh đặc trưng, 48120 kết nối.
- Lớp $F6$ có 84 nơ ron, 10164 tham số. Hàm truyền

$$a_i = f(n_i) = A tanh(Sn_i) \tag{3.5}$$

  $A = 1.7159$.
- Lớp đầu ra (10×1)

$$y_i = \sum_j (x_j - w_{i,j})^2 \tag{3.6}$$

3.3.2 Các lớp trong mạng tích chập

- Ảnh đầu vào.
  - Ảnh 2 chiều (2-D): $w \times h \times c$, trong đó w là độ rộng, h là độ dài và c là số kênh của ảnh. $c = 1$-ảnh đen trắng, $c = 3$ - ảnh mầu.
  - Ảnh 3 chiều (3-D)$w \times h \times \times d \times c$, trong đó w là độ rộng, h là độ dài, d là độ sâu và c là số kênh của ảnh.
  - Ví dụ ảnh 2-D:

  moon = imread('moon.tif'); imshow(moon);

Hình 3.7: Ảnh 2-D, độ phân giải 537×358

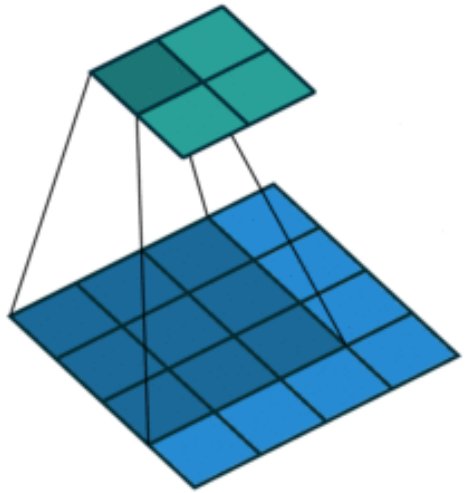
3.3.2 Các lớp trong mạng tích chập

```
>> moon(100:105,200:205)

ans =

  6×6 uint8 matrix

   207   209   209   207   214   215
   204   207   205   209   217   221
   209   213   209   214   217   224
   212   206   213   211   211   218
   215   216   212   209   216   215
   216   225   211   210   224   214
```

Hình 3.8: Một số điểm ảnh của hình mặt trăng

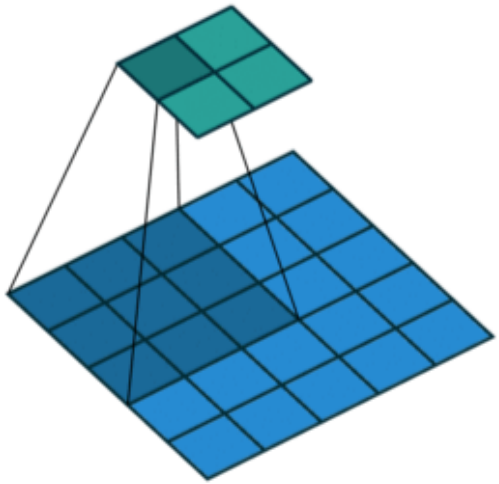
3.3.2 Các lớp trong mạng tích chập

- Lớp tích chập
  - Đầu vào: $P_{m \times n}$
  - $W_{f \times f}$- ma trận trọng số của nơ-ron, còn được gọi là bộ lọc có kích thước là $f < min(m, n)$.
  - Đầu ra: $A = P * W$.



Hình 3.9: *Stride* $= 1$

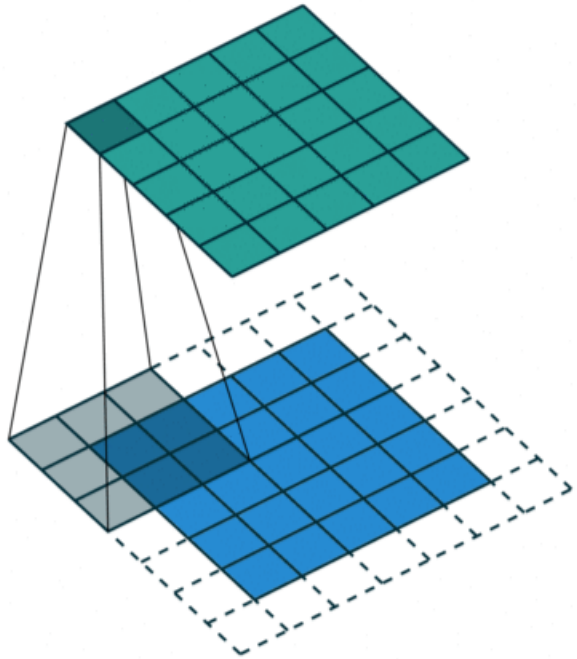
a) Lớp tích chập



Hình 3.10: *Stride* $= 2$

a) Lớp tích chập

Padding = 1. Có thể thay đổi kích thước đầu ra của lớp.



Hình 3.11: *Padding* = [1 1]

b) Lớp chuẩn hóa

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{3.7}$$

μ_B-giá trị trung bình, σ_B^2 - phương sai (mini batch, mỗi kênh)

$$y_i = \gamma \hat{x}_i + \beta \tag{3.8}$$

γ, β - các tham số học

c) Lớp ReLu

Hàm Relu:

$$\begin{aligned} f(x) &= x, \text{nếu } x \geq 0 \\ &= 0, \text{nếu } x < 0 \end{aligned} \tag{3.9}$$

Hàm LeakyRelu:

$$\begin{aligned} f(x) &= x, \text{nếu } x \geq 0 \\ &= \alpha x, \text{nếu } x < 0, \alpha = 0.01 \end{aligned} \tag{3.10}$$

d) Lớp pooling

- Làm giảm kích thước đầu vào.
- Chia đầu vào thành các vùng có dạng hình chữ nhật và lấy giá trị trung bình hoặc cực đại mỗi vùng.

- Hàm max
- Hàm average

e) Lớp softmax

- Hàm softmax.

$$y_r(x) = \frac{e^{a_j(x)}}{\sum_{j=1}^{k} e^{a_j(x)}} \tag{3.11}$$

$a_j(x)$ - là đầu vào thứ j.

3.3.3 Nhận dạng chữ viết tay

- Lớp đầu vào:
  $input\_layer = imageInputLayer(inputSize)$

$$inputSize = [h \quad w \quad c], \tag{3.12}$$

h - độ cao, w - độ rộng, c - số kênh
$c = 1$ ảnh đen trắng
$c = 3$ ảnh màu.

3.3.3 Nhận dạng chữ viết tay

- Lớp tích chập:
  $conv\_layer = convolution2dLayer(filterSize, numFilters)$

$$\begin{aligned} filterSize &= [h \quad w], \\ numFilters &= \text{ số filter} \end{aligned} \tag{3.13}$$

  h - độ cao, w - độ rộng, c - số kênh
- $conv\_layer = convolution2dLayer(filterSize, numFilters, 'P1', V1, 'P2', V2, ...)$.
- 'Stride': $[u \quad v]$, u - bước dọc, v - bước ngang
- 'Padding': 'same', 'scalar', [a b], [t b l r].

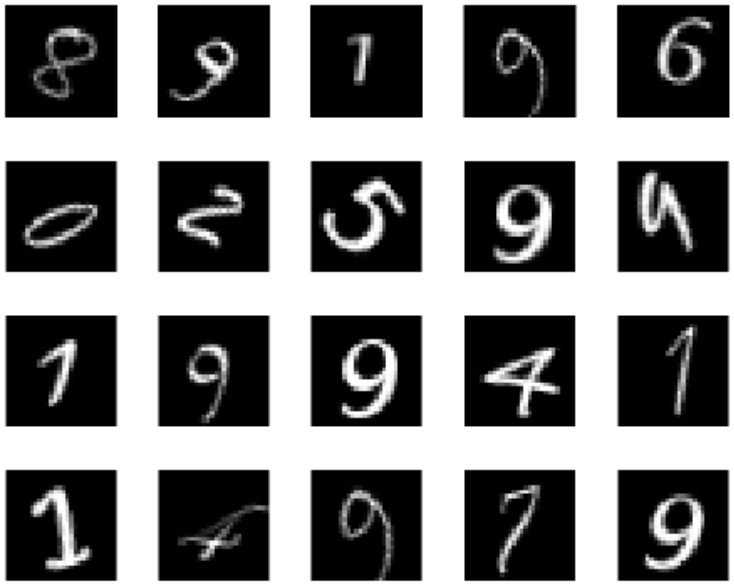
3.3.3 Nhận dạng chữ viết tay

Các lớp trong Matlab:

- batchNormalizationLayer
- reluLayer
- fullyConnectedLayer(outputSize)
- softmaxLayer
- classificationLayer

Ví dụ nhận dạng 10 chữ số viết tay

- Nhận dạng 10 chữ số viết tay $0 \div 9$.
- Tập mẫu gồm 10000 ảnh, mỗi chữ số có 1000 ảnh mẫu.



Hình 3.12: Một số ảnh chữ số viết tay

```
digitDatasetPath = ...
    fullfile(matlabroot,'toolbox','nnet','nndemos', ...
    'nndatasets','DigitDataset');
imds = imageDatastore(digitDatasetPath, ...
    'IncludeSubfolders',true,'LabelSource','foldernames');
figure;
perm = randperm(10000,20);
for i = 1:20
    subplot(4,5,i);
    imshow(imds.Files{perm(i)});
end
labelCount = countEachLabel(imds)
img = readimage(imds,1);
size(img)
numTrainFiles = 750;
[imdsTrain,imdsValidation] = ...
    splitEachLabel(imds,numTrainFiles,'randomize');
layers = [
    imageInputLayer([28 28 1])
    convolution2dLayer(3,8,'Padding','same')
    batchNormalizationLayer
```

```
20      reluLayer
21      maxPooling2dLayer(2,'Stride',2)
22      convolution2dLayer(3,16,'Padding','same')
23      batchNormalizationLayer
24      reluLayer
25      maxPooling2dLayer(2,'Stride',2)
26      convolution2dLayer(3,32,'Padding','same')
27      batchNormalizationLayer
28      reluLayer
29      fullyConnectedLayer(10)
30      softmaxLayer
31      classificationLayer];
32  options = trainingOptions('rmsprop', ...
33      'InitialLearnRate',0.01, ...
34      'MaxEpochs',4, ...
35      'Shuffle','every-epoch', ...
36      'ValidationData',imdsValidation, ...
37      'ValidationFrequency',30, ...
38      'Verbose',false, ...
39      'Plots','training-progress',...
40      'ExecutionEnvironment','cpu');
41  net = trainNetwork(imdsTrain,layers,options);
```

```
42 YPred = classify(net,imdsValidation);
43 YValidation = imdsValidation.Labels;
44 accuracy = sum(YPred == YValidation)/numel(YValidation)
```

3.4. Một số vấn đề trong mạng CNN

- 3.4.1 Mạng nhiều lớp trong học sâu
- 3.4.2 Vanishing gradient

3.4.1 Mạng nhiều lớp trong học sâu

- Mạng perceptron: Bài toán đường biên tuyến tính
- Mạng hai lớp: Bài toán xấp xỉ
- Mạng nhiều lớp: Xử lý ảnh, video, giọng nói

3.4.2 Vanishing gradient

- Quá trình lan truyền thuận

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^{m} + \mathbf{b}^{m+1}), \forall\, m = 0, 1, ..., M-1. \tag{3.14}$$

$$\mathbf{a}^{0} = \mathbf{p}, \mathbf{a} = \mathbf{a}^{M}. \tag{3.15}$$

- Độ nhạy các lớp đầu vào càng bé khi số lớp càng lớn

$$\mathbf{s}^{M} = -2\dot{\mathbf{F}}^{M}(\mathbf{n}^{M})(\mathbf{t} - \mathbf{a}), \tag{3.16}$$

$$\mathbf{s}^{m} = \dot{\mathbf{F}}^{m}(\mathbf{n}^{m})(\mathbf{W}^{m+1})^{T}\mathbf{s}^{m+1}, \forall\, m = M-1, ..., 2, 1. \tag{3.17}$$

- Quá trình cập nhật các tham số của mạng (gradient các lớp đầu vào bé)

$$\mathbf{W}^{m}(k+1) = \mathbf{W}^{m}(k) - \alpha\mathbf{s}^{m}(\mathbf{a}^{m-1})^{T} \tag{3.18}$$

$$\mathbf{b}^{m}(k+1) = \mathbf{b}^{m}(k) - \alpha\mathbf{s}^{m} \tag{3.19}$$

3.5 Một số mạng CNN

- Dữ liệu ImageNet[2]
- Một số mạng CNN tiêu biểu

[2] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015

Dữ liệu ImageNet[3]

Dữ liệu cuộc thi ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

- 1000 đối tượng khác nhau.
- 1.281.167 ảnh cho huấn luyện mạng.
- $50,000$ ảnh để đánh giá mô hình (validation).
- 100000 ảnh để kiểm tra (test).
- Huấn luyện các mô hình nhận diện đối tượng và phân loại.

[3] https://www.image-net.org/download.php

Một số mạng CNN tiêu biểu

- GoogLeNet.
  - Có 144 lớp (sâu 22 lớp), có thể phân loại 1000 đối tượng khác nhau.
  - Kích thước ảnh đầu vào là $224 \times 224 \times 3$.
  - Hàm crossentropyex

$$J = -\sum_{i=1}^{Q}\sum_{j=1}^{K} t_{ij} ln y_{ij}, \tag{3.20}$$

Q - số mẫu, K - số lớp cần phân loại, t_{ij} - đầu ra mẫu tương ứng với mẫu thứ i thuộc loại j, y_{ij} là đầu ra thứ j của lớp *softmax* tương ứng với mẫu thứ i.

Một số mạng CNN tiêu biểu

- VGG16.
  - 42 lớp (sâu 16 lớp).
  - Phân loại 1000 đối tượng khác nhau.
  - Kích thước ảnh đầu vào là $224 \times 224 \times 3$.
  - Hàm crossentropyex

Một số mạng CNN tiêu biểu

- Mạng Alexnet.
  - Có khả năng phân loại 1000 đối tượng khác nhau.
  - 25 lớp (sâu 8 lớp).
  - Ảnh đầu vào có kích thước $227 \times 227 \times 3$.
  - Hàm crossentropyex

Một số mạng CNN tiêu biểu

Network	Depth	Size	Parameters (Millions)	Image Input Size
squeezenet	18	5.2 MB	1.24	227-by-227
googlenet	22	27 MB	7.0	224-by-224
inceptionv3	48	89 MB	23.9	299-by-299
densenet201	201	77 MB	20.0	224-by-224
mobilenetv2	53	13 MB	3.5	224-by-224
resnet18	18	44 MB	11.7	224-by-224
resnet50	50	96 MB	25.6	224-by-224
resnet101	101	167 MB	44.6	224-by-224
xception	71	85 MB	22.9	299-by-299
inceptionresnetv2	164	209 MB	55.9	299-by-299
shufflenet	50	5.4 MB	1.4	224-by-224
nasnetmobile	*	20 MB	5.3	224-by-224
nasnetlarge	*	332 MB	88.9	331-by-331
darknet19	19	78 MB	20.8	256-by-256
darknet53	53	155 MB	41.6	256-by-256
efficientnetb0	82	20 MB	5.3	224-by-224
alexnet	8	227 MB	61.0	227-by-227
vgg16	16	515 MB	138	224-by-224
vgg19	19	535 MB	144	224-by-224

Hình 3.13: Các mạng sâu[4]

[4]Deep Learning Toolbox User's Guide. Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth

3.6. Ứng dụng của CNN trong phân loại và nhận dạng

- Nhận dạng khuôn mặt

Hình 3.14: Ảnh khuôn mặt.

3.6. Ứng dụng của CNN trong phân loại và nhận dạng

Phân loại ảnh ung thư.

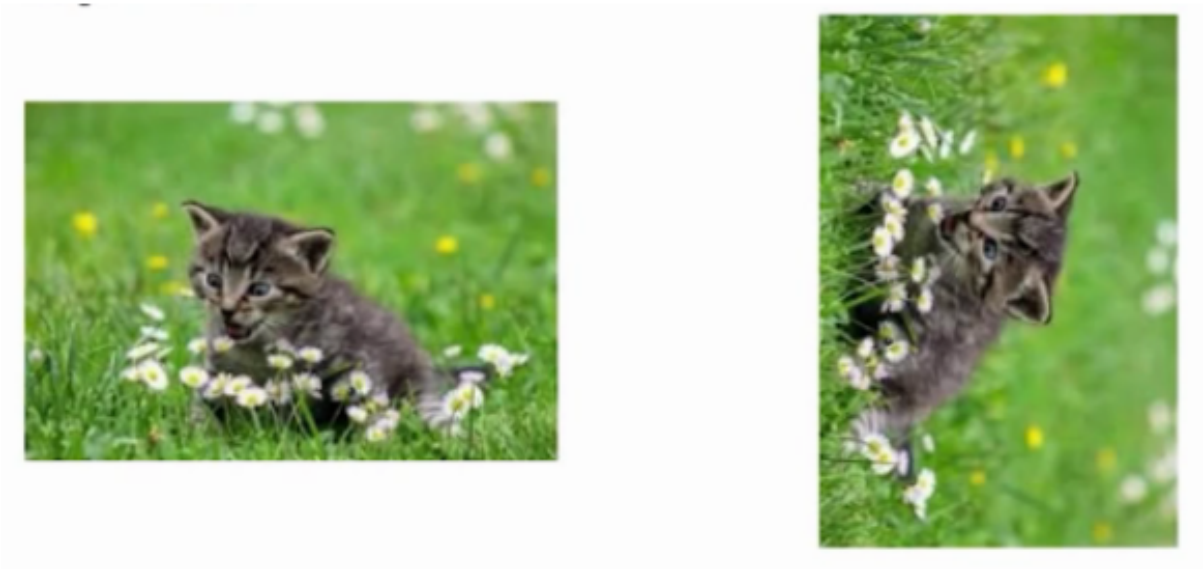
- Hình ảnh chụp X quang và mô bệnh học (MRI-Magnetic Resonance Imaging, X-ray-Ionizing radiation, CT-Computed Tomography, US - Ultrasound, PET-Positron Emission Tomography).
- Phân loại ảnh, tái tạo ảnh, dò ảnh, phân vùng ảnh.
- Các mô hình đã được huấn luận (pretrained models).

3.7. Một số kỹ thuật trong nhận dạng dùng CNN

- 3.7.1. Gia tăng dữ liệu
- 3.7.2. Học chuyển đổi

3.7.1. Gia tăng dữ liệu

$$B = imrotate(A, angle)$$



Hình 3.15: Xoay ảnh

3.7.1. Gia tăng dữ liệu

$B = imtranslate(A, Translation)$


Hình 3.16: Dịch ảnh

$$\begin{aligned} X &= x + \delta x \\ Y &= y + \delta y \end{aligned} \tag{3.21}$$

3.7.1. Gia tăng dữ liệu

$$B = flip(A, dim)$$

Hình 3.17: Flipping

3.7.1. Gia tăng dữ liệu

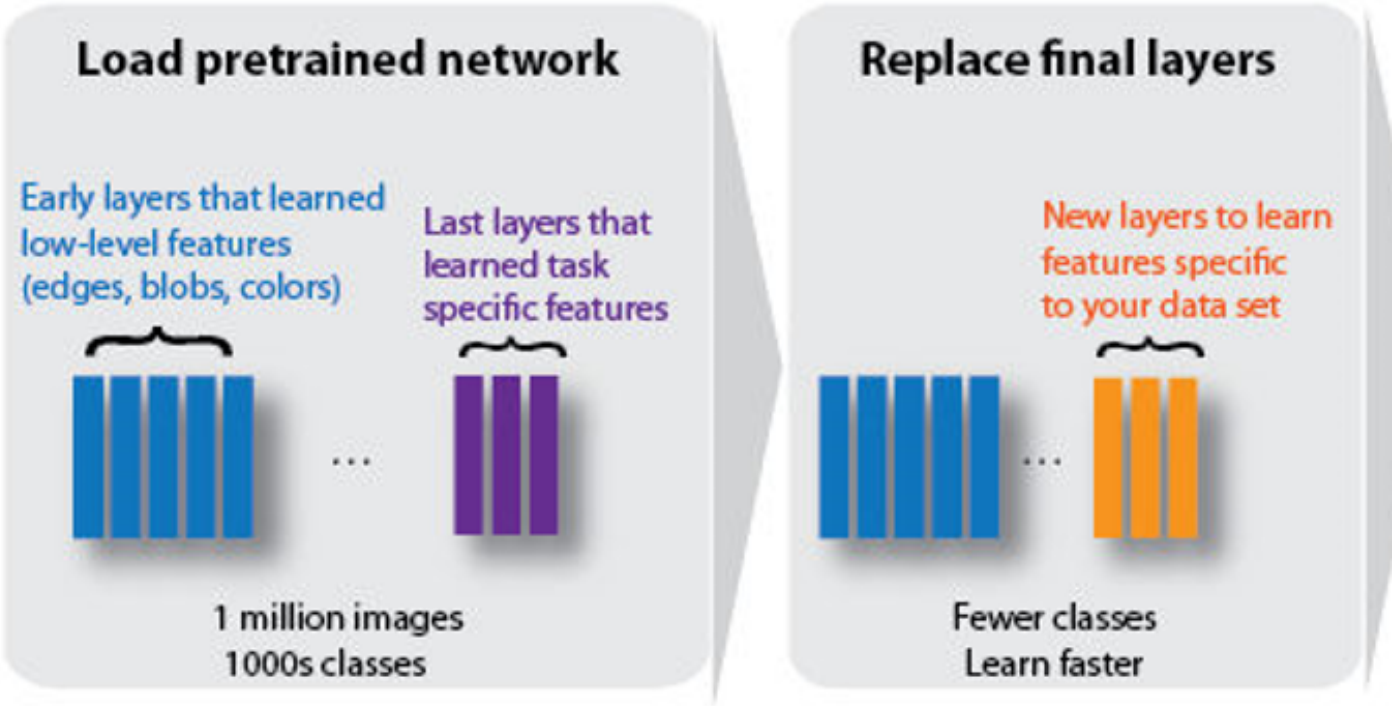
$$B = imnoise(A, Type)$$

Hình 3.18: Làm nhiễu ảnh

3.7.1. Gia tăng dữ liệu

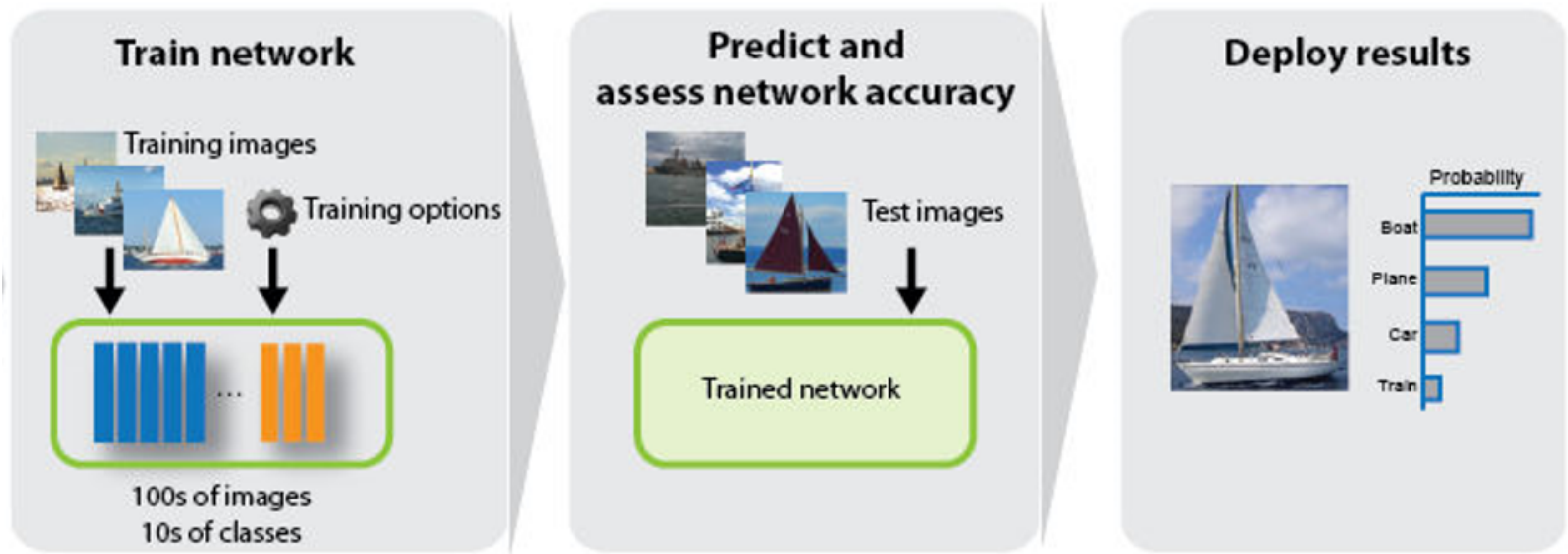
Hình 3.19: Làm mờ ảnh

3.7.2. Học chuyển đổi



Hình 3.20: Mạng đã học và mạng mới

3.7.2. Học chuyển đổi



Hình 3.21: Huấn luyện mạng và thử nghiệm

Phương pháp gradient ngẫu nhiên - Stochastic Gradient Descent (SGD)

- Cho hàm mục tiêu $F(\mathbf{x})$, trong đó $\mathbf{x}$ là véc tơ các tham số (trọng số, ngưỡng). Tìm nghiệm tối ưu $\mathbf{x}^*$ sao cho $F(\mathbf{x}^*) \rightarrow min$.
- Công thức lặp

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F, \tag{3.22}$$

trong đó $\mathbf{x}_k$ là véc tơ tham số hiện tại, $\mathbf{x}_{k+1}$ là véc tơ tham số mới ở lần lặp thứ k, $\nabla F = \frac{\partial F}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_k}$ là véc tơ gradient, và $\alpha > 0$ là tốc độ học.

Phương pháp SGD với hệ số chỉnh hướng học[5]

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F + \gamma(\mathbf{x}_k - \mathbf{x}_{k-1}), \tag{3.23}$$

trong đó $\gamma > 0$ là hệ số chỉnh hướng học.

[5] [2] Murphy, K. P. Machine Learning: A Probabilistic Perspective. The MIT Press, Cambridge, Massachusetts, 2012.

Phương pháp RMSProp

$$v_k = \beta_2 v_{k-1} + (1-\beta_2)(\nabla F)^T \nabla F \tag{3.24}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\alpha \nabla F}{\sqrt{v_k} + \epsilon}, \tag{3.25}$$

Phương pháp Adam[6] - Adaptive moment estimation

$$m_k = \beta_1 m_{k-1} + (1-\beta_1)\nabla F \tag{3.26}$$

$$v_k = \beta_2 v_{k-1} + (1-\beta_2)(\nabla F)^T \nabla F \tag{3.27}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\alpha m_k}{\sqrt{v_k} + \epsilon}, \tag{3.28}$$

[6]Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).