Deep Learning Methods for Image Segmentation Containing Translucent Overlapped Objects

Tayebeh Lotfi Mahyari, Richard M. Dansereau

Systems and Computer Engineering, Carleton University

Ottawa, Canada

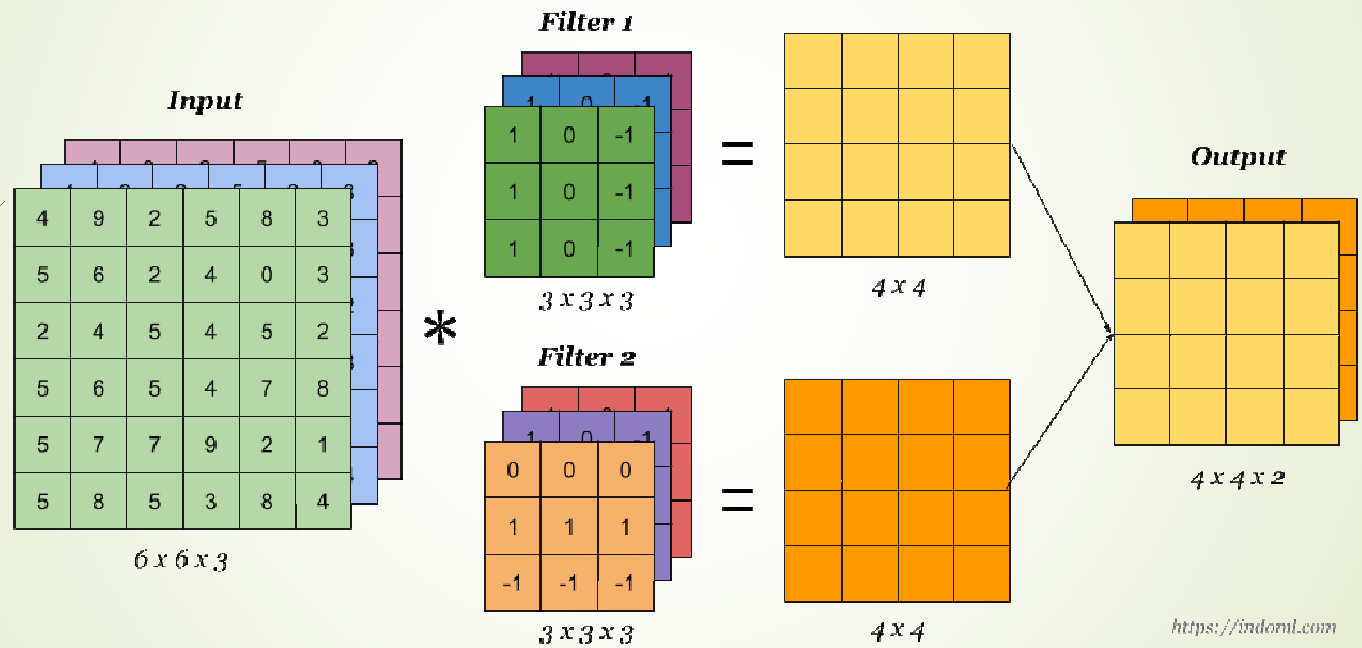
Outline

- Convolutional neural networks (CNNs)
- Image classification and image segmentation on CNNs
  - AlexNet, VGGNet, GoogLeNet, ResNet
  - SegNet
- Proposed network
  - Transfer learning
  - Proposed network
  - Proposed residual network
- Results
- Conclusion

Convolutional Neural Networks(CNNs)

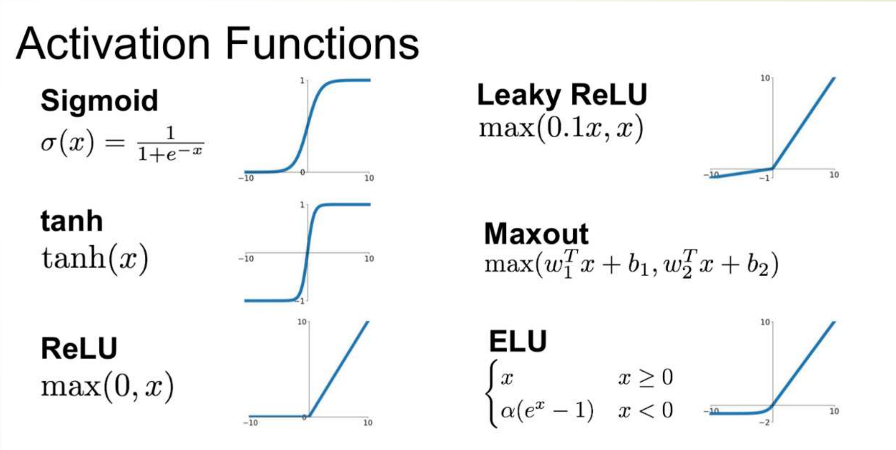
- Deep learning in machine learning (2006)
- CNNs
  - Convolutional layers
  - Activation layers
  - Batch normalization layers
  - Pooling layers
  - Fully-connected layers

Convolutional Layers



Activation Layers

- Rectified linear unit
- Leaky ReLU
- Tanh
- Sigmoid
- Maxout
- ELU



Batch Normalization Layers

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

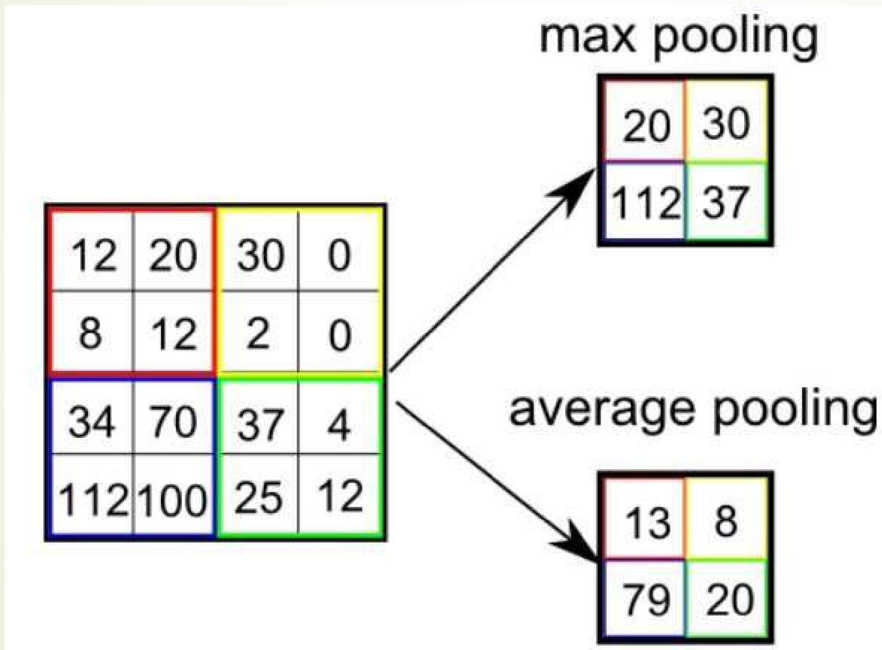
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Pooling Layers



Fully-connected Layers

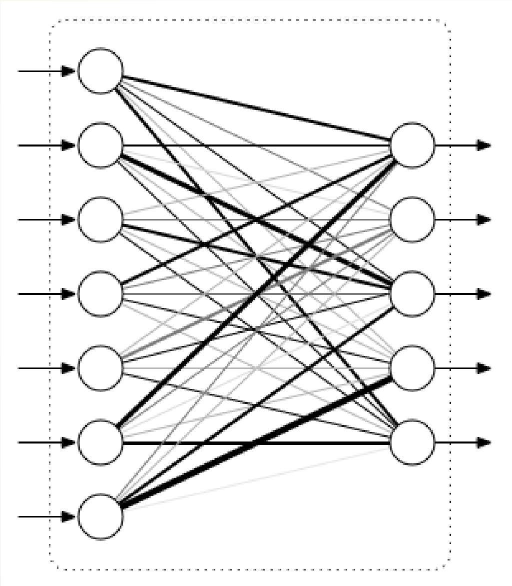
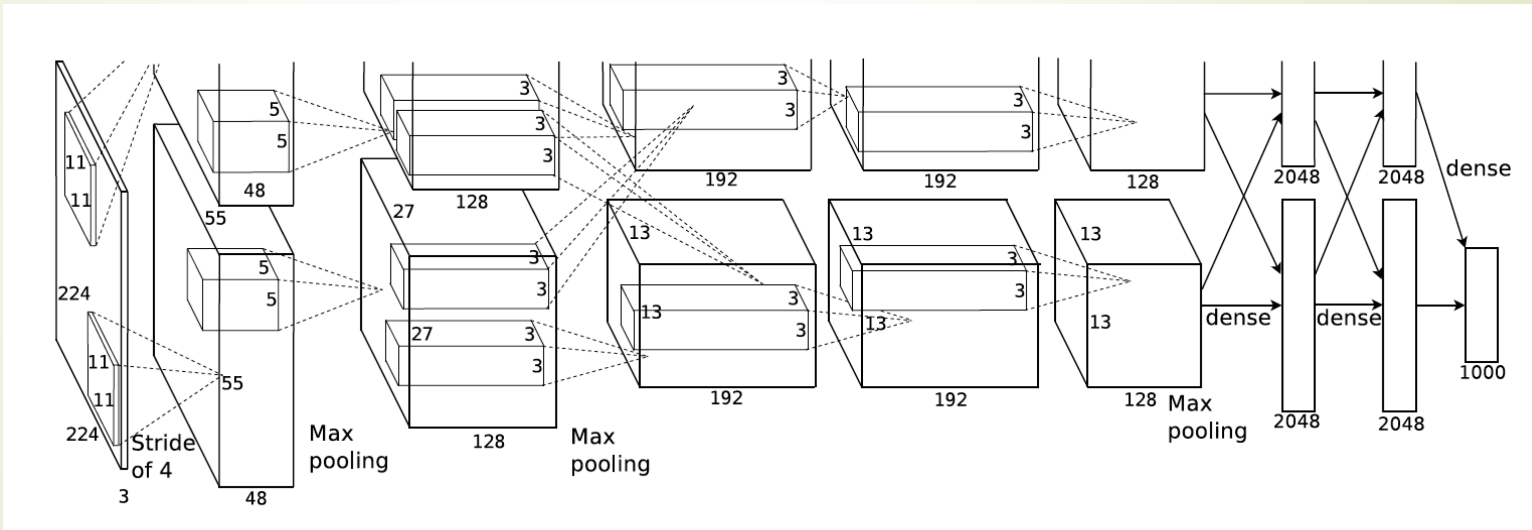


Image Classification & Image Segmentation

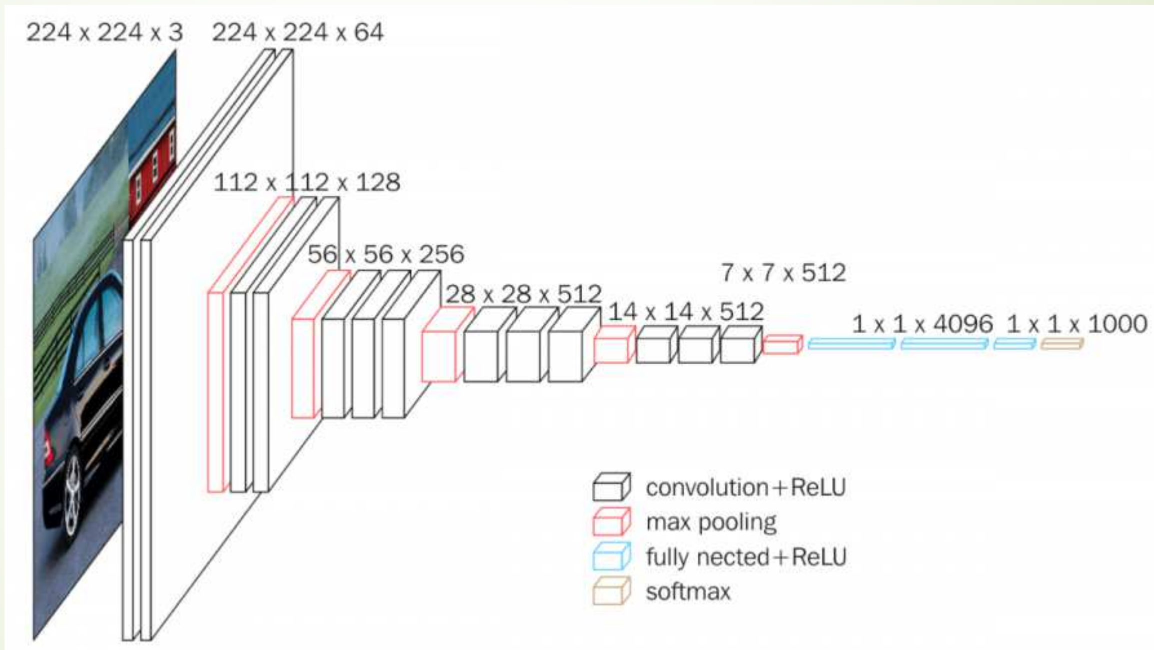
- AlexNet [1]
- VGGNet [2]
- GoogLeNet [3]
- ResNet[4]
- SegNet [5]

AlexNet

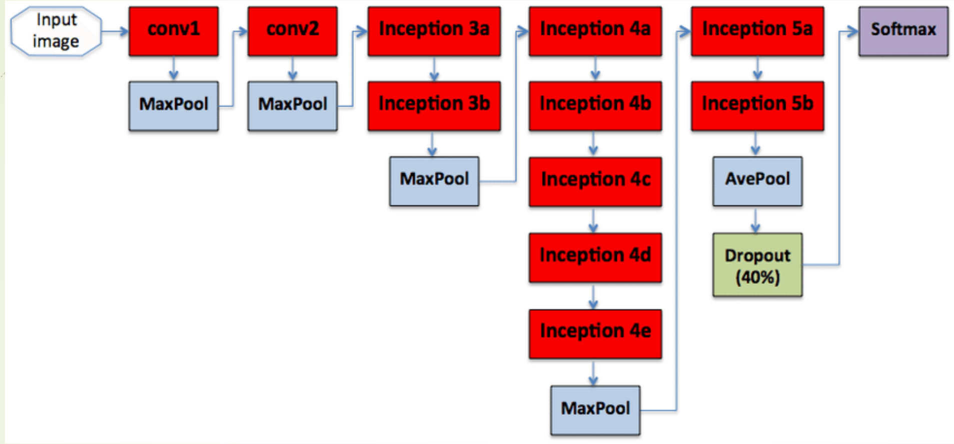


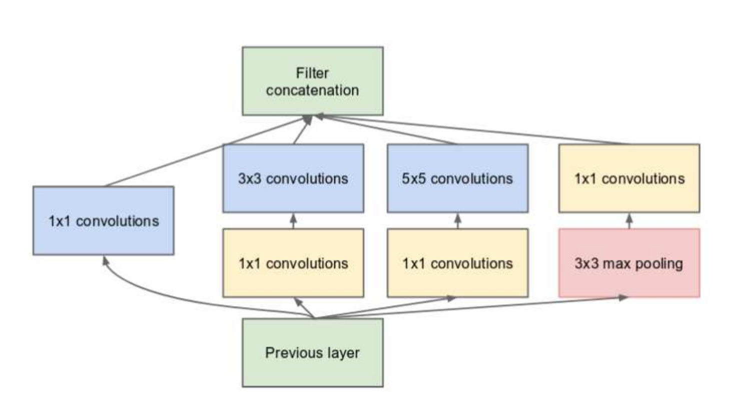
224
224
11
11
11
11
Stride of 4
3
55
55
5
5
5
5
48
48
Max pooling
27
27
3
3
3
3
3
128
128
Max pooling
13
13
3
3
192
192
13
13
3
3
192
192
13
13
128
128
Max pooling
dense
dense
dense
2048
2048
2048
2048
1000

VGGNet


224 x 224 x 3
224 x 224 x 64
112 x 112 x 128
56 x 56 x 256
28 x 28 x 512
14 x 14 x 512
7 x 7 x 512
1 x 1 x 4096
1 x 1 x 1000
convolution+ReLU
max pooling
fully nected+ReLU
softmax

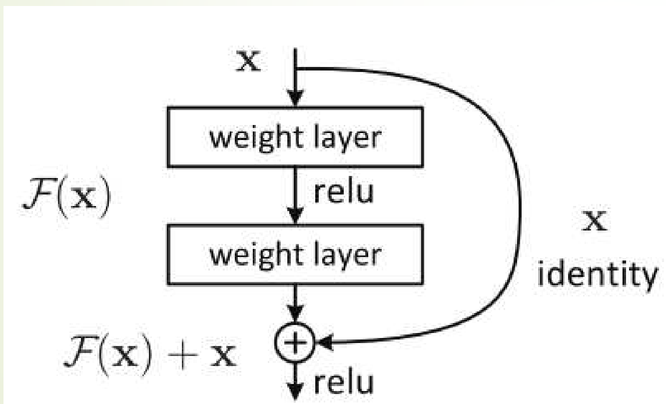
GoogleNet



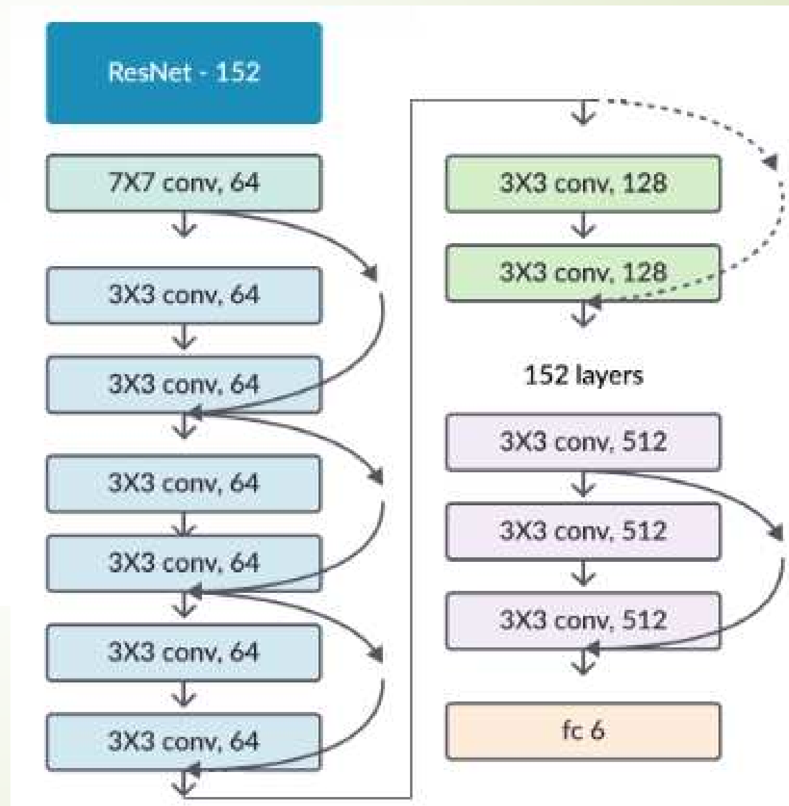


(b) Inception module with dimension reductions

ResNet

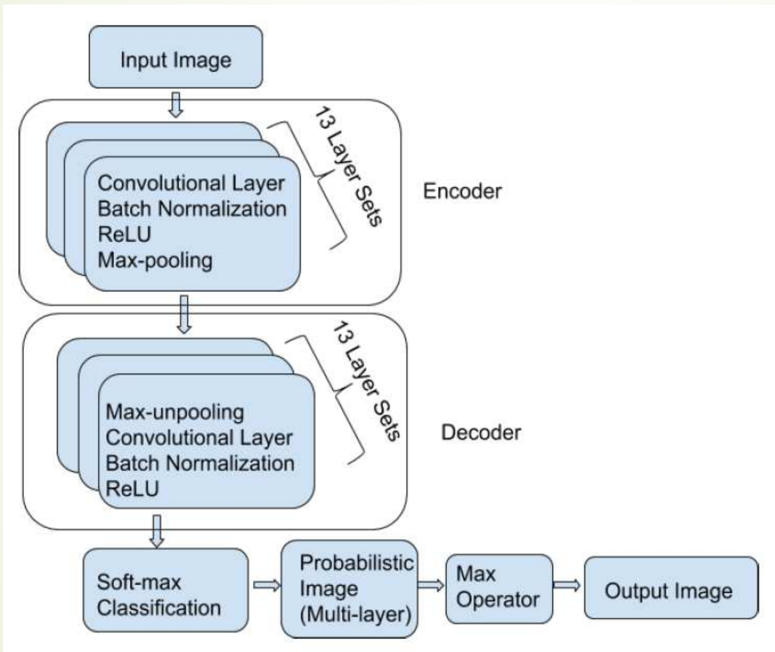


x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu
x
identity



ResNet - 152
7X7 conv, 64
3X3 conv, 64
3X3 conv, 64
3X3 conv, 64
3X3 conv, 64
3X3 conv, 64
3X3 conv, 64
3X3 conv, 128
3X3 conv, 128
152 layers
3X3 conv, 512
3X3 conv, 512
3X3 conv, 512
fc 6

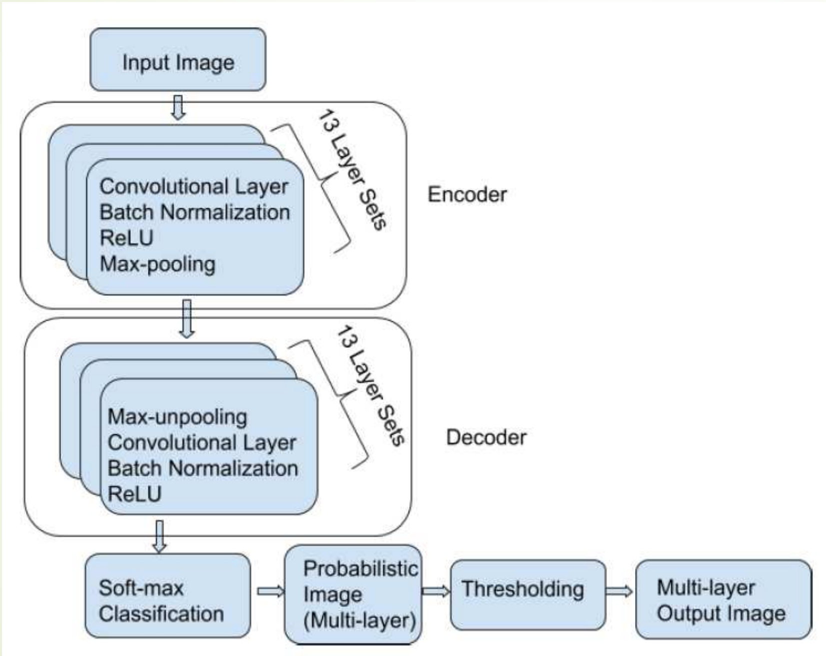
SegNet



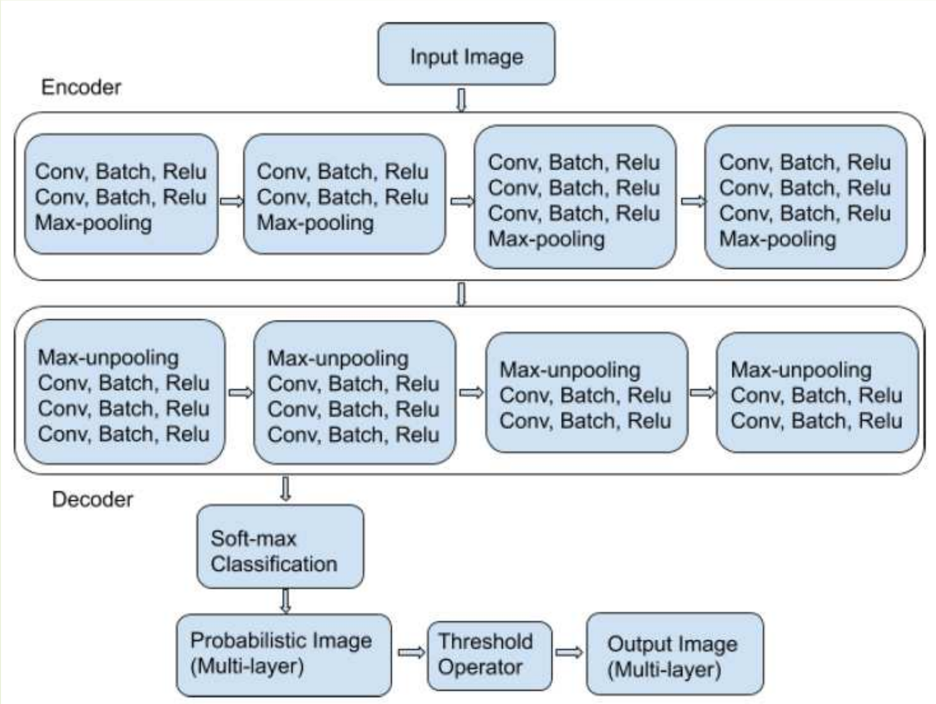
Input Image
13 Layer Sets
Convolutional Layer
Batch Normalization
ReLU
Max-pooling
Encoder
13 Layer Sets
Max-unpooling
Convolutional Layer
Batch Normalization
ReLU
Decoder
Soft-max Classification
Probabilistic Image (Multi-layer)
Max Operator
Output Image

CNNs for Segmenting Translucent Partly Overlapped Objects

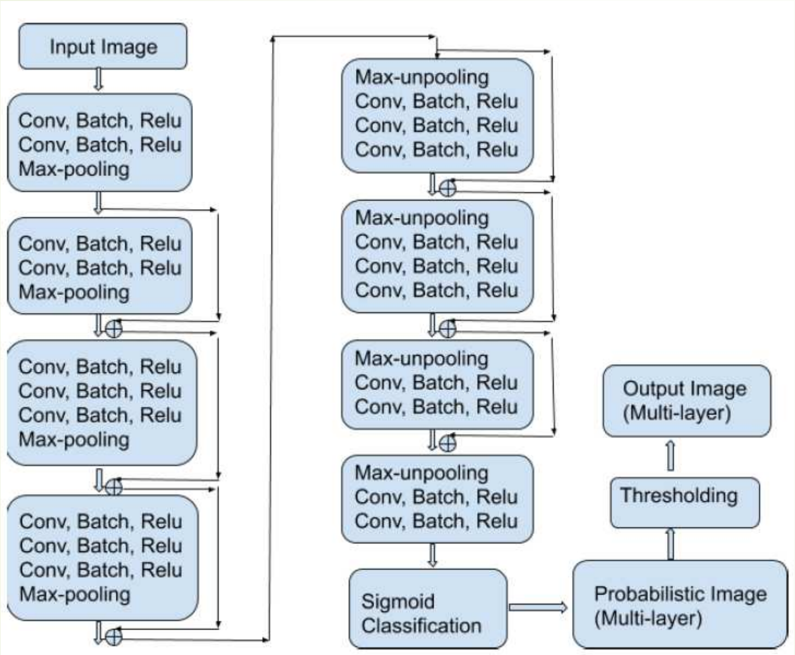
Transfer Learning



Proposed Network

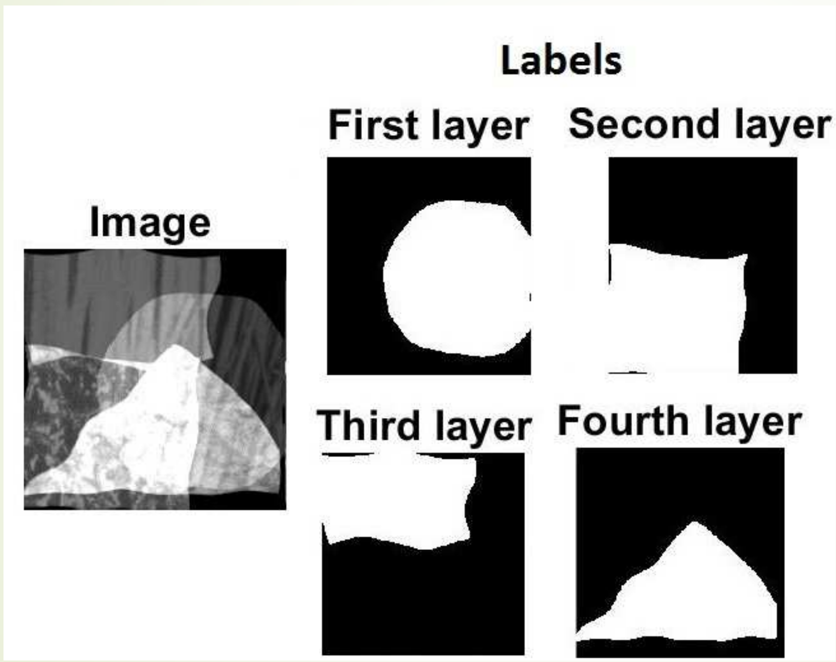


Proposed Residual Network



Results

Datasets1



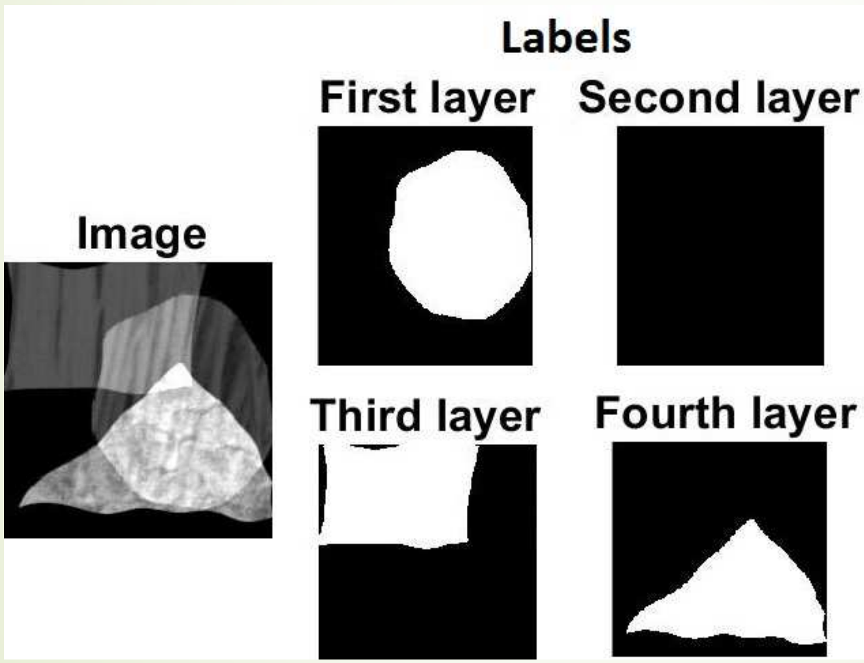
$$I = \sum_{m=1}^{4} a_{m,l} * \{(O_m * P_m(x,y)) \circledast G\},$$

$$P_m(x,y) = P_m(x : x+V, y : y+W),$$

$$1 < x, \quad x+V < S, \quad 1 < y, \quad y+W < T.$$

$$a_{m,l} = a_m + N_l(0,\sigma), \qquad N_l = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{l^2}{2\sigma^2}}$$

Dataset2



$$I = \sum_{m=1}^{4} a_{m,l} * \{(O_m * P_m(x, y)) \circledast G\},$$

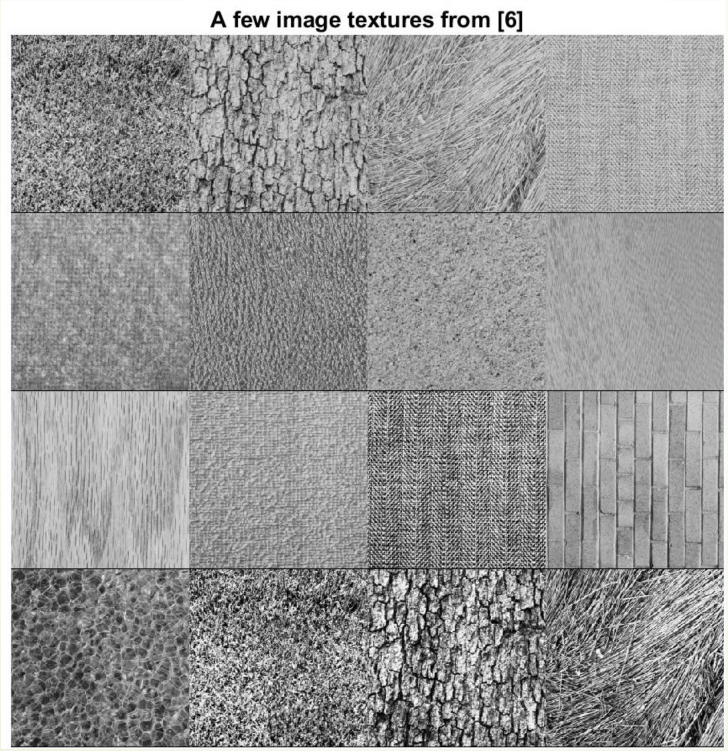
$$P_m(x, y) = P_m(x : x + V, y : y + W),$$

$$1 < x, \quad x + V < S, \quad 1 < y, \quad y + W < T.$$

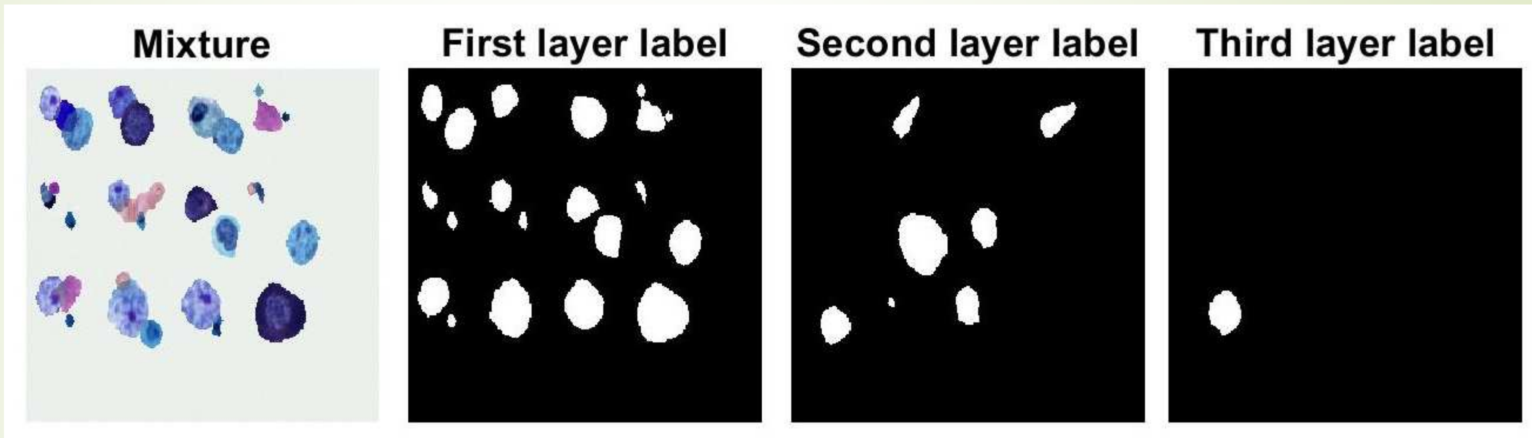
$$a_{m,l} = \begin{cases} a_m + N_l(0, \sigma), & \text{if the component is selected} \\ 0, & \text{otherwise} \end{cases}$$

$$N_l = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{l^2}{2\sigma^2}}$$

Examples of Image Textures



Dataset3



$$F(p) = Ap + q$$

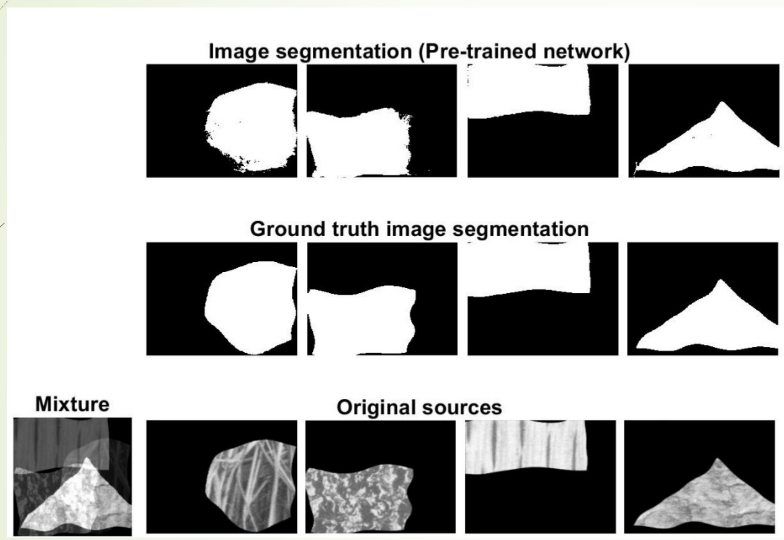
$$I_i = \sum_{n=1}^{3} R_{in} + \sum_{n \neq p, n, p=1}^{3} \alpha_n J_{inp} + \sum_{n=1}^{3} \beta_n K_{in}$$

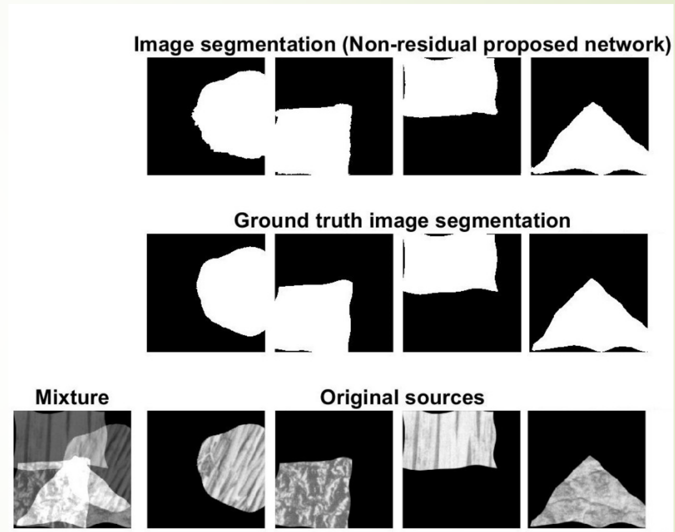
$$J_{inp} = \{Q_i | Q_i = O_{in} \cap O_{ip}, \quad O_{in} \in I_{in}, \quad O_{ip} \in I_{ip}, \quad n \neq p, \quad n, p = 1, 2, 3\}$$

$$K_i n = \{Q_i | Q_i = \cap_{n=1}^{3} O_{in} \quad O_{in} \in I_{in}\}$$

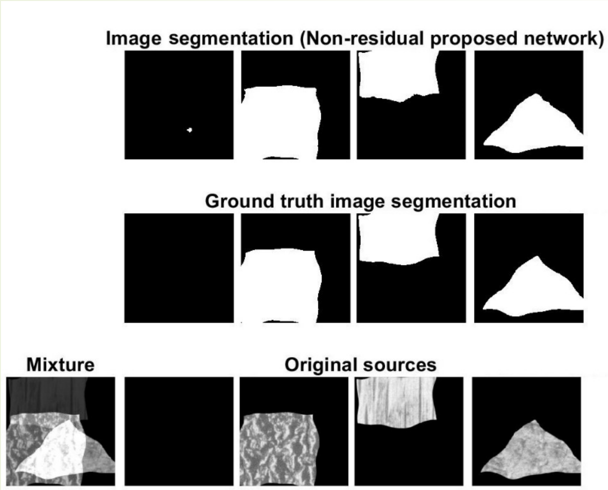
$$\alpha_n = \begin{cases} 0.45 & \text{for } n = 1, \\ 0.35 & \text{for } n = 2, \\ 0.45 & \text{for } n = 3, \end{cases} \qquad \beta_n = \begin{cases} 0.2 & \text{for } n = 1, \\ 0.12 & \text{for } n = 2, \\ 0.2 & \text{for } n = 3. \end{cases}$$

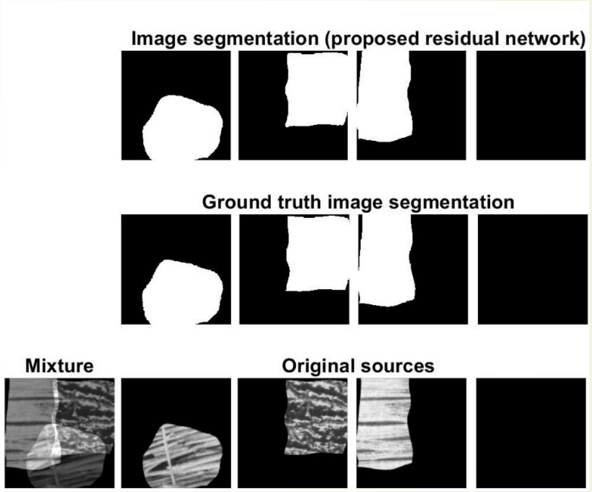
Results for Dataset1



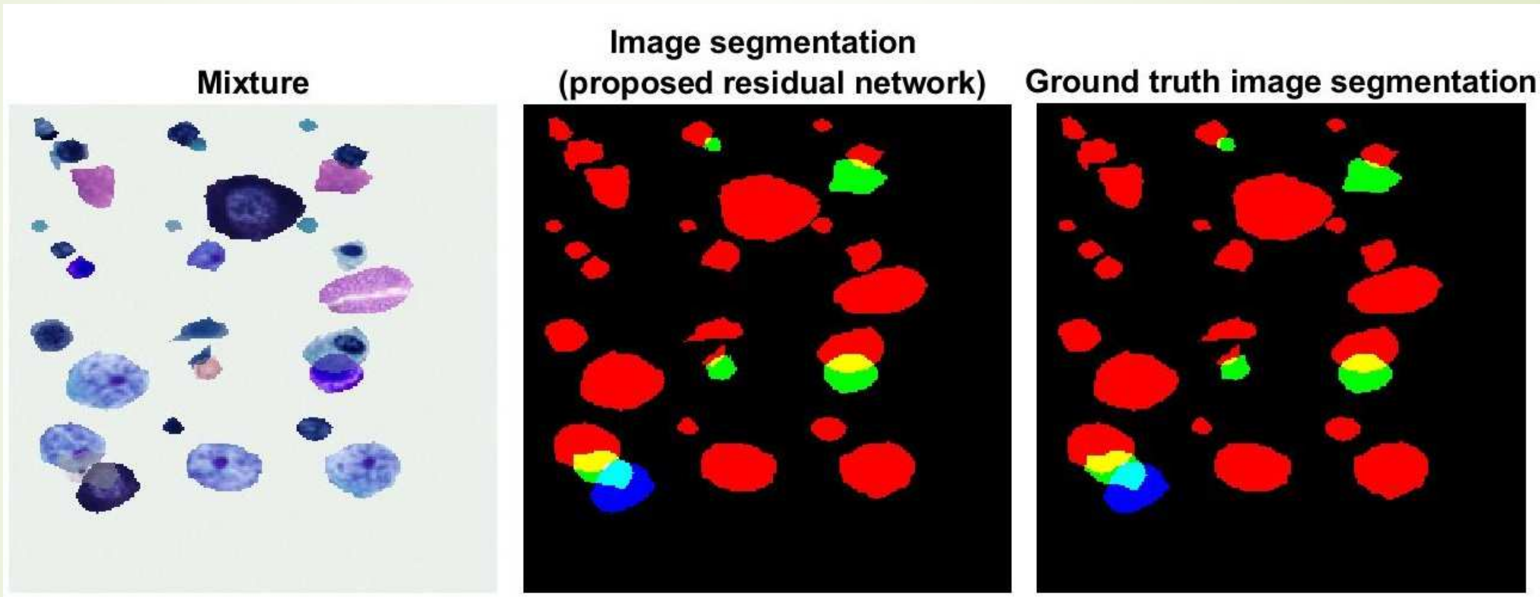


Results for Dataset2





Results for Dataset3



Segmentation Results

- $Acc = \frac{|\Sigma_G = \Sigma_R|}{|V|} \times 100$
- $IoU = \frac{\Sigma_G \cap \Sigma_R}{\Sigma_G \cup \Sigma_R} \times 100$

Description	Accuracy (%)	IoU (%)	Time (Sec)
Transfer_First	98.14	95.13	13425
Proposed1_First	99.06	97.48	2907
Proposed1_Second	99.37	97.59	3043
Proposed2_Second	99.55	98.07	829
Proposed2_Third	99.78	95.81	101320

Conclusion & Future Work

- CNNs for image segmentation using partially-overlapped translucent objects
- Pre-trained network for transfer learning (SegNet)
- New non-residual network
- New residual network
- Applying our residual network on real data.

Questions

References

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012.
2. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.
3. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.
4. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CoRR, vol. abs/1512.03385, 2015.
5. V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. CoRR, abs/1505.07293, 2015.
6. USC texture database. http://sipi.usc.edu/database/database.php?volume=textures.