

Convolutional Neural Networks

David Dagan
HARVARD MEDICAL SCHOOL | BRIGHAM AND WOMEN'S HOSPITAL | DANNA-FARBBER CANCER INSTITUTE
SAM Joint Imaging Therapy Scientific Symposium Certificate Series Session 5:
Convolutional Neural Nets - Wednesday, 8/1/2018 1:45 PM - 3:45 PM

Why do we need convolutions?

CNN specifics

CNN flavors

Resources

Why do we need convolutions?

CNN specifics

CNN flavors

Resources

TensorFlow - MNIST For Beginners

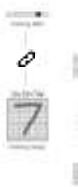


https://www.tensorflow.org/tutorials/mnist_for_beginners

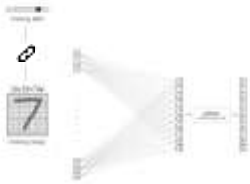
TensorFlow - MNIST For Beginners



TensorFlow - MNIST For Beginners



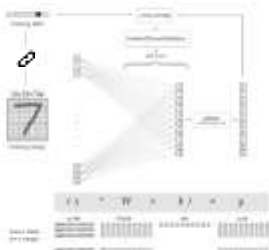
TensorFlow - MNIST For Beginners



TensorFlow - MNIST For Beginners



TensorFlow - MNIST For Beginners



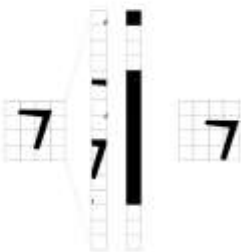
Translation Variance



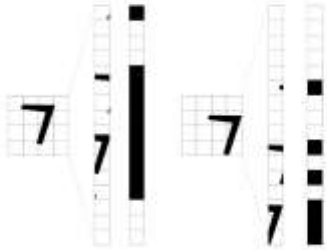
Translation Variance



Translation Variance



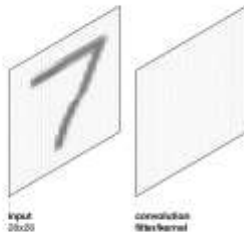
Translation Variance



Weight Sharing



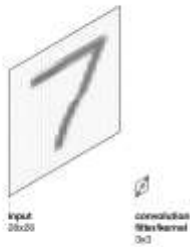
Convolutions



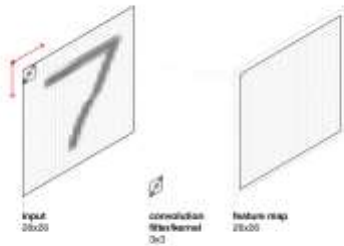
Convolutions



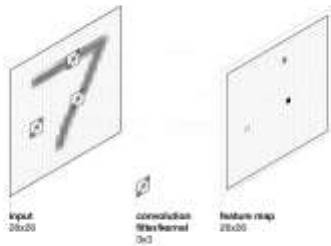
Convolutions



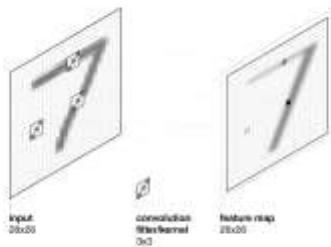
Convolutions



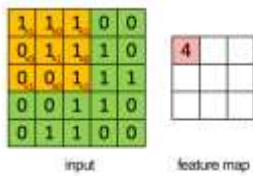
Convolutions



Convolutions



Convolutions

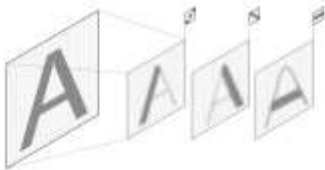


http://yqjzhu.com/2018/08/1/convolution-operations.html

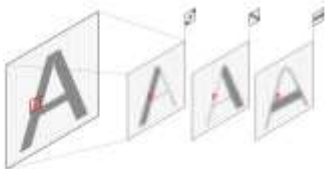
Convolutions



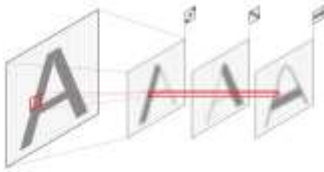
Convolutions



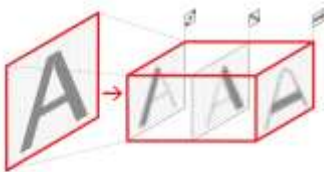
Convolutions



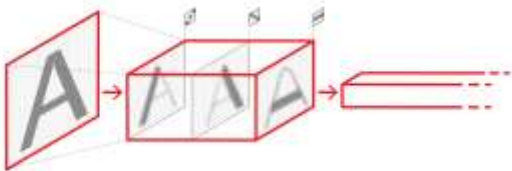
Convolutions



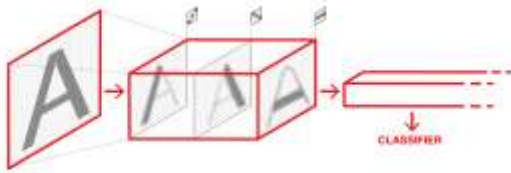
Convolutions



Convolutions



Convolutions



Why do we need convolutions?

CNN specifics

Convolution

Feature Maps

Neocognitron

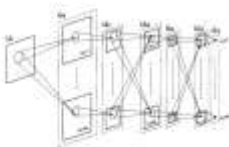


Fig. 1. Schematic diagram of the Neocognitron.



Fig. 2. Illustration showing the basic mechanism of the cell when a target cell fires.

Kunihiko Kodaira
Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Utilized by Sali in Primate
Biological Cybernetics - 1980

LeNet-5

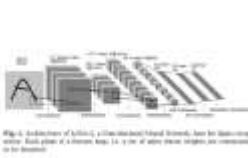


Fig. 3. Samples of inputs, outputs, and feature maps in LeNet-5. The gray boxes in the upper row correspond to specific filters in the input layer.

From Lec 11: Deep Neural Networks: Applications of Convolution
Object Recognition with Gradient Based Learning: Steps, Context and Grouping in ConvNet Tutorials - 1999

AlexNet



Figure 4: Architecture of AlexNet. AlexNet is a deep convolutional neural network (CNN) designed to run on two GPUs in parallel. It consists of five layers: an input layer of size 224x224x3, two convolutional layers of size 128x128x48 and 128x128x128, two fully connected layers of size 2048x2048 and 2048x1000, and an output layer of size 1000. AlexNet is trained using stochastic gradient descent with momentum and dropout.

See Introduction: The History of AlexNet by Jeff Dean
ImageNet Classification with Deep Convolutional Neural Networks
Advances in Neural Information Processing Systems - 2012

AlexNet @ ImageNet



Figure 5: Architecture of AlexNet. AlexNet is a deep convolutional neural network (CNN) designed to run on two GPUs in parallel. It consists of five layers: an input layer of size 224x224x3, two convolutional layers of size 128x128x48 and 128x128x128, two fully connected layers of size 2048x2048 and 2048x1000, and an output layer of size 1000. AlexNet is trained using stochastic gradient descent with momentum and dropout.



From: AlexNet on the ImageNet dataset, by Alex Krizhevsky et al.

Hyperparameters

of filters
32, 64, 128 ...



Photo by [shutterstock.com](#)

Hyperparameters

filter size
11x11, 5x5, 3x3 ...



Hyperparameters

padding
pad the input image?

stride
of pixels to shift the filter

Hyperparameters



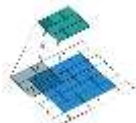
padding: no
stride: 2

<http://graphics.stanford.edu/~csdz/vis/conv2d.html>

Hyperparameters



padding: no
stride: 2



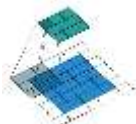
padding: yes
stride: 2

<http://graphics.stanford.edu/~csdz/vis/conv2d.html>

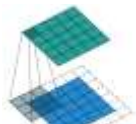
Hyperparameters



padding: no
stride: 2



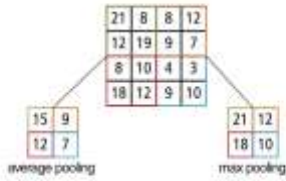
padding: yes
stride: 2



padding: yes
stride: 1

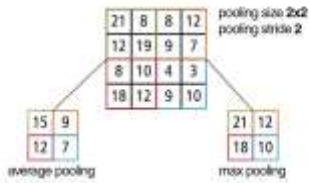
<http://graphics.stanford.edu/~csdz/vis/conv2d.html>

Pooling



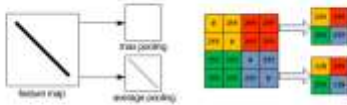
<http://www.cs.toronto.edu/~rsb/tutorial/notes/convolution/convolution.html#pooling>

Pooling



<http://www.cs.toronto.edu/~rsb/tutorial/notes/convolution/convolution.html#pooling>

Pooling



Deep Learning: From Theory to Practice
 Max Pooling for Convolutional Neural Networks
 International Conference on Image and Knowledge Technology - 2014

Pooling

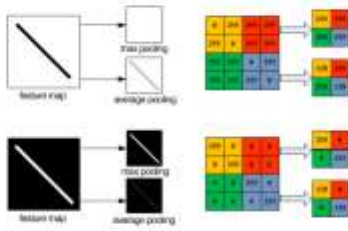


Figure 10.10 Pooling. From: *Deep Learning*.
 Adapted: *Practical Deep Learning for Convolutional Neural Networks*.
 International Conference on Intelligent Systems and Knowledge Technology - 2014

Convolutional Neural Networks

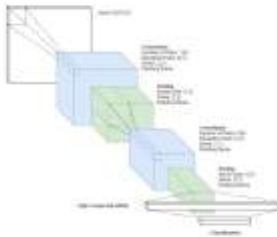


Figure 10.11 Convolutional Neural Network. From: *Deep Learning*.
 Adapted: *Practical Deep Learning for Convolutional Neural Networks*.
 International Conference on Intelligent Systems and Knowledge Technology - 2014

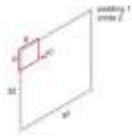
Data Augmentation

- rotate
- translate
- shear
- flip
- scale/zoom
- crop
- apply whitening
- apply noise
- shift channel
- shift brightness/contrast
- blur

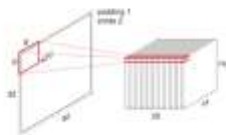


Figure 10.12 Data Augmentation. From: *Deep Learning*.
 Adapted: *Practical Deep Learning for Convolutional Neural Networks*.
 International Conference on Intelligent Systems and Knowledge Technology - 2014

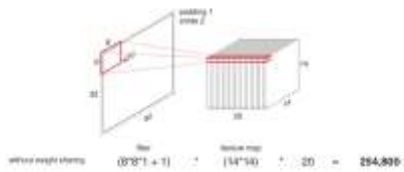
Number of Parameters



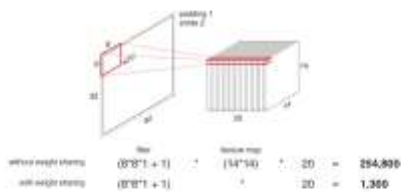
Number of Parameters



Number of Parameters



Number of Parameters



Memory Management

BNF1 (1024x64)	memory	24 1024x64	weights	0
CONV1a (128x128x64)	memory	24 128x128x64	weights	17 17x17x4 = 128
CONV1b (128x128x64)	memory	24 128x128x64	weights	17 17x17x4 = 128
POOL1 (128x128x32)	memory	12 128x128x32	weights	0
CONV2a (128x128x32)	memory	12 128x128x32	weights	17 17x17x3 = 128
CONV2b (128x128x32)	memory	12 128x128x32	weights	17 17x17x3 = 128
POOL2 (128x128x16)	memory	6 128x128x16	weights	0
CONV3a (128x128x16)	memory	6 128x128x16	weights	17 17x17x3 = 128
CONV3b (128x128x16)	memory	6 128x128x16	weights	17 17x17x3 = 128
POOL3 (128x128x8)	memory	3 128x128x8	weights	0
CONV4a (128x128x8)	memory	3 128x128x8	weights	17 17x17x3 = 128
CONV4b (128x128x8)	memory	3 128x128x8	weights	17 17x17x3 = 128
POOL4 (128x128x4)	memory	1 128x128x4	weights	0
FC_1a (4096)	memory	4096	weights	4096x4096 = 16,777,216
FC_1b (4096)	memory	4096	weights	4096x4096 = 16,777,216

17x17 kernel (128x128x64) = 128

From University of Waterloo Research

Very Deep Convolutional Networks for Large-Scale Image Recognition
 International Conference on Image Syn and Knowledge Technology - 2014

Memory Management

BNF1 (1024x64)	memory	24 1024x64	weights	0
CONV1a (128x128x64)	memory	24 128x128x64	weights	17 17x17x4 = 128
CONV1b (128x128x64)	memory	24 128x128x64	weights	17 17x17x4 = 128
POOL1 (128x128x32)	memory	12 128x128x32	weights	0
CONV2a (128x128x32)	memory	12 128x128x32	weights	17 17x17x3 = 128
CONV2b (128x128x32)	memory	12 128x128x32	weights	17 17x17x3 = 128
POOL2 (128x128x16)	memory	6 128x128x16	weights	0
CONV3a (128x128x16)	memory	6 128x128x16	weights	17 17x17x3 = 128
CONV3b (128x128x16)	memory	6 128x128x16	weights	17 17x17x3 = 128
POOL3 (128x128x8)	memory	3 128x128x8	weights	0
CONV4a (128x128x8)	memory	3 128x128x8	weights	17 17x17x3 = 128
CONV4b (128x128x8)	memory	3 128x128x8	weights	17 17x17x3 = 128
POOL4 (128x128x4)	memory	1 128x128x4	weights	0
FC_1a (4096)	memory	4096	weights	4096x4096 = 16,777,216
FC_1b (4096)	memory	4096	weights	4096x4096 = 16,777,216

17x17 kernel (128x128x64) = 128

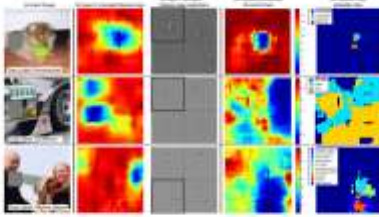
From University of Waterloo Research

Very Deep Convolutional Networks for Large-Scale Image Recognition
 International Conference on Image Syn and Knowledge Technology - 2014



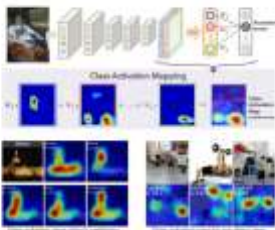
Mellin GeForce GTX 980 4GB

Visualizing Attention



Jonathan Shlens et al. 2016
 Visualizing and Understanding Convolutional Networks
 International Conference on Computer Vision - 2014

Visualizing Attention



Bolei Zhou, Abhinav Khosla, Yuxin Li, Jiaxun Tang, Han Xiao, Liang-Yan Xiao, and Philip H. Torr
 Learning Deep Features for Discriminative Localization
 Conference on Computer Vision and Pattern Recognition - 2014

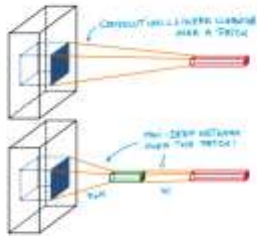
Why do we need attention mechanisms?

CNN architectures

CNN flavors

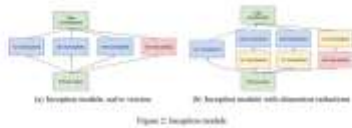
Feedback loop

1x1 Convolutions



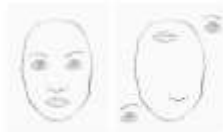
1x1 Convolutions
 Udacity Deep Learning Nanodegree Program

Inception Module



Udacity Deep Learning Nanodegree Program
 Going Deeper with Convolutions (GoogleNet/Inception)
 CS784, 2015

Capsule Networks



http://robotics.wisc.edu/~dave/robotics/capsule-networks/capsule-networks-2016-11-16

Capsule Networks



http://arxiv.org/pdf/1608.01247v1.pdf

Capsule Networks

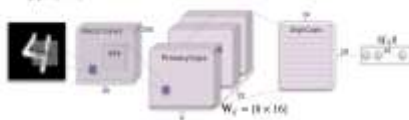


"The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster." - Geoffrey Hinton

http://arxiv.org/pdf/1608.01247v1.pdf

Capsule Networks

Figure 1: A single CapsNet with 3 layers. This model gives comparable results to deep convolutional networks (such as AlexNet and VGG) [13][14]. The length of the activity vector of each capsule in DigitCaps layer indicates presence of an instance of each class and is used to calculate the classification loss. W_{ij} is a weight matrix between each $u_i \in \mathbb{R}^3, (i = 1, 2, 3)$ in PrimaryCapsules and $v_{ij} \in \mathbb{R}^3, (j = 1, 10)$.



See slides: Static View of DigitCaps
Dynamic Routing Between Capsules
Conference on Neural Information Processing Systems - 2017

ResNets

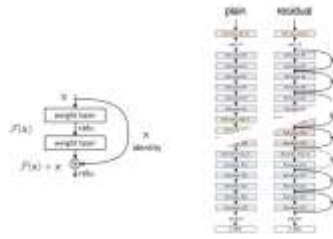


Image: Kim, Wenyang, Donggyu, Kim, et al. *Deep Residual Learning for Image Recognition. Conference on Computer Vision and Pattern Recognition, 2016*



Figure 1: Residual Networks are conceptually shown as skip connections in Equation (1). When we expand the summation in Equation (1), we obtain an unrolled view of a 1-block residual network. In a 1-block residual network, each path contains a $\text{Conv}(\cdot)$ operation. From this view, it is apparent that residual networks have $O(2^k)$ implicit paths connecting input and output and that adding a block doubles the number of paths.

Image: Kim, Wenyang, Donggyu, et al. *Residual Networks Behave Like Ensembles of Relatively Shallow Networks. Conference on Computer Vision and Pattern Recognition, 2016*

Fully Convolutional Networks

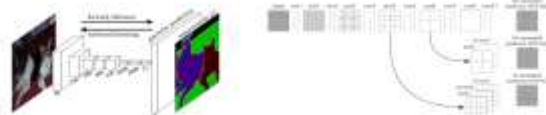


Figure 1: Fully convolutional networks and affinity maps. Fully convolutional networks (FCNs) are used to produce pixel-wise semantic segmentation.

Image: Long, Jianbo, Shelhamer, et al. *Fully Convolutional Networks for Semantic Segmentation. CVPR, 2015*

Fully Convolutional Networks



Fig. 1. U-Net architecture for medical image segmentation. The encoder path is a multi-scale feature map. The number of layers is shown on top of the box. The decoder is parallel on the lower left side of the box. When feature maps are concatenated, the network learns the different operations.

102 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
 U-Net: Convolutional Neural Networks for Biomedical Image Segmentation
 MICCAI 2015

Fully Convolutional Networks



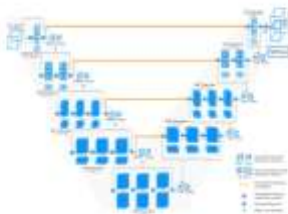
Fig. 1. U-Net architecture for medical image segmentation. The encoder path is a multi-scale feature map. The number of layers is shown on top of the box. The decoder is parallel on the lower left side of the box. When feature maps are concatenated, the network learns the different operations.



Fig. 2. Finding the intensity for medical segmentation of cellular segmentation that represents the maximum likelihood of the binary threshold of the segmentation on the output of a convolutional layer with the 3x3 kernel. Finding the best threshold is implemented by thresholding.

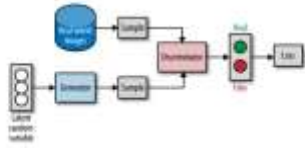
102 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
 U-Net: Convolutional Neural Networks for Biomedical Image Segmentation
 MICCAI 2015

Fully Convolutional Networks



From: Medical Image Segmentation: A Survey
 V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation
<https://arxiv.org/pdf/1606.04787v2>

Generative Adversarial Networks (GAN)



<http://www.cs.toronto.edu/~rfw/papers/2014-gan.pdf>

Generative Adversarial Networks (GAN)

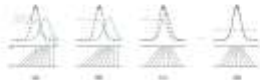


Figure 1. Evolution of the distribution of generated images over time. (a) Initial distribution of generated images, which is very narrow and centered differently from the target distribution. (b) Distribution after 1000 iterations. (c) Distribution after 10000 iterations. (d) Distribution after 100000 iterations. The target distribution is shown as a solid line. The generated distribution is shown as a shaded area.

arXiv:1410.5026v1 [cs.LG] 26 Oct 2014
Generative Adversarial Nets
 Advances in Neural Information Processing Systems - 2014

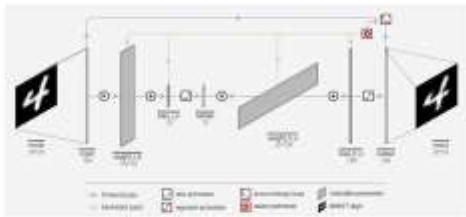
Deep Convolutional Generative Adversarial Networks (DCGAN)



Figure 1. DCGAN generator used for CIFAR-10 image synthesis. A 100-dimensional vector z is sampled from a standard normal distribution and fed into the generator. A series of four sequentially stacked convolutional layers create feature maps that are sampled without activation and then convolved with high-pass filters to produce noise. Finally, an MLP convolutional pooling layer is used.

arXiv:1410.5026v1 [cs.LG] 26 Oct 2014
Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
 International Conference on Learning Representations - 2014

Variational Autoencoders



Why do we need a distribution?

CNN Architecture

Latent Variable

Resources

Transfer Learning



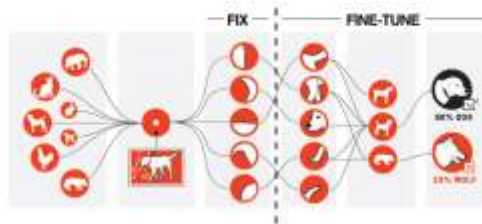
For more information, please refer to the lecture.

Transfer Learning



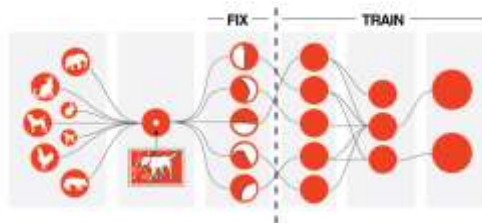
For (https://www.kaggle.com/abhinavkumar1994/transfer-learning)

Transfer Learning



For (https://www.kaggle.com/abhinavkumar1994/transfer-learning)

Transfer Learning



For (https://www.kaggle.com/abhinavkumar1994/transfer-learning)

Transfer Learning



<https://www.microsoft.com/en-us/research/blog/transfer-learning/>

Transfer Learning

Year	Year	Author(s)	Topic	Topicality	No. of Publications
2000	2000-2002	Yan Lecun et al			62 references
2012	2009-2011	Josiah Davis, Tom Mitchell	Image classification	100%	62 references
2012	2010-2011	Andrew Senior, Richard Schaefer	Text classification	100%	62 references
2012	2009-2011	David Rosenberg	Image classification	100%	62 references
2012	2009-2011	David Rosenberg	Image classification	100%	62 references
2012	2009-2011	David Rosenberg	Image classification	100%	62 references

<https://www.microsoft.com/en-us/research/blog/transfer-learning-progress-report-2012-2013/>

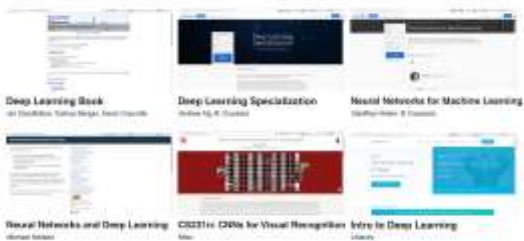
Online Resources



Online Resources



Online Resources



Reproducibility

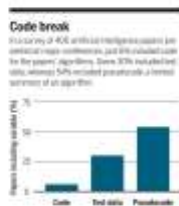


Goodfellow, Ian J., et al. "A Survey on Deep Learning in Medical Image Analysis." *Medical Image Analysis*. 2017.

Deep Learning

Open-Source Deep Learning Tech
github.com

Reproducibility



Sherry Chen
Artificial Intelligence Faces Reproducibility Crisis
Science 2018

Existing Solutions

DeepFace: Face Recognition with DeepFace
Caffe: Convolutional Architecture for Fast Feature Embedding
arxiv.org/pdf/1401.2959

Existing Solutions



Existing Medical Imaging Solutions



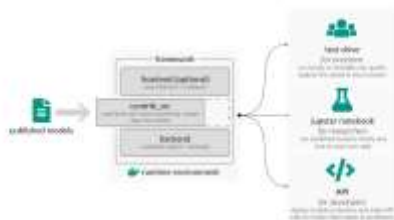


Components



Source: [ModEdu: Model-Driven Analysis of Software and Usage \(MDS\) - Amy ModEdu: Plug & Predict Solutions for Reproducible AI Research](#)
 modEdu.d

How it Works



Source: [ModEdu: Model-Driven Analysis of Software and Usage \(MDS\) - Amy ModEdu: Plug & Predict Solutions for Reproducible AI Research](#)
 modEdu.d

For Contributors



Source: [ModEdu: Model-Driven Analysis of Software and Usage \(MDS\) - Amy ModEdu: Plug & Predict Solutions for Reproducible AI Research](#)
 modEdu.d

For Contributors



ModelHub: Model Serving, Analytics, Feature and Deployment
ModelHub: Plug & Predict Solutions for Reproducible AI Research
modelhub.ai

For Contributors



ModelHub: Model Serving, Analytics, Feature and Deployment
ModelHub: Plug & Predict Solutions for Reproducible AI Research
modelhub.ai

For Contributors



ModelHub: Model Serving, Analytics, Feature and Deployment
ModelHub: Plug & Predict Solutions for Reproducible AI Research
modelhub.ai



Community Outreach



co-authorship through model contributions

ModelHub: Michael Schmitt, Andrey Y. Zhurav and Filippos Panagiotou
 ModelHub: Plug & Predict Solutions for Reproducible AI Research
 modelhub.ai