E1 216 COMPUTER VISION

LECTURE 12: LEARNING IN VISION

Venu Madhav Govindu
Department of Electrical Engineering
Indian Institute of Science, Bengaluru

2022

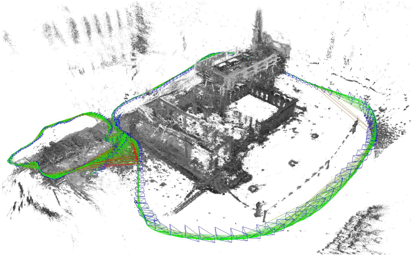
- Why do we need **learning** in vision?
- Should every solution be **learnt**?

- Why do we need **learning** in vision?
- Should every solution be **learnt**?

Tasks in Computer Vision

- Segmentation, Recognition, Detection, Localisation
- Tasks on the image plane $\mathbb{R}^2$
- Deep Learning breakthrough, with problems

Adapted from Fig. 6.1 in Szeliski *Computer Vision: Algorithms and Applications*, draft 2nd edition

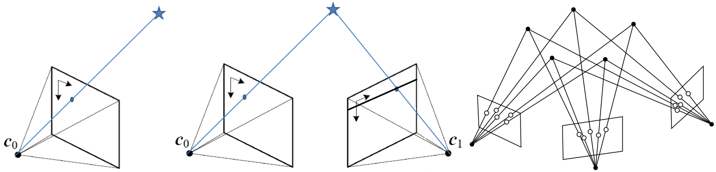


Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.

Dense models of several landmarks produced by COLMAP's MVS pipeline.

Geometric Problems in Computer Vision

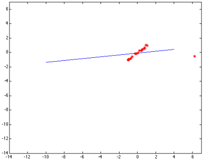
- 3D Reconstruction from multiple images
- Geometry induced by pinhole camera
- Reasoning about 3D world from 2D images
- **Explicit** reasoning and engineering used

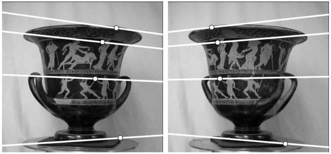


Geometric Models are Explicit

- Geometric relations governed by pinhole model
- **Explicit** models for observations
- Epipolar Geometry: $\hat{\mathbf{x}}^T F \mathbf{x} = 0$
- Reprojection Error can be written in **explicit** form

Learning in Vision





Role of Learning

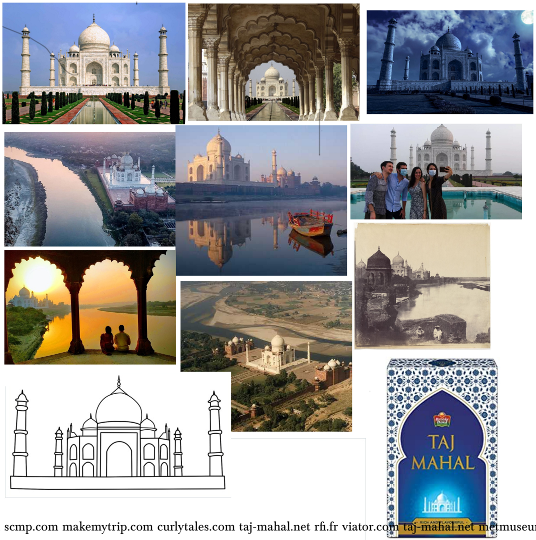
- Different types of tasks
    - Motion Estimation
    - Shape Analysis
    - Segmentation
- Theory, Model, Algorithms
- Understanding of physics (geometry) and statistics
- Higher-level Reasoning?

Learning in Vision

Learning in Vision

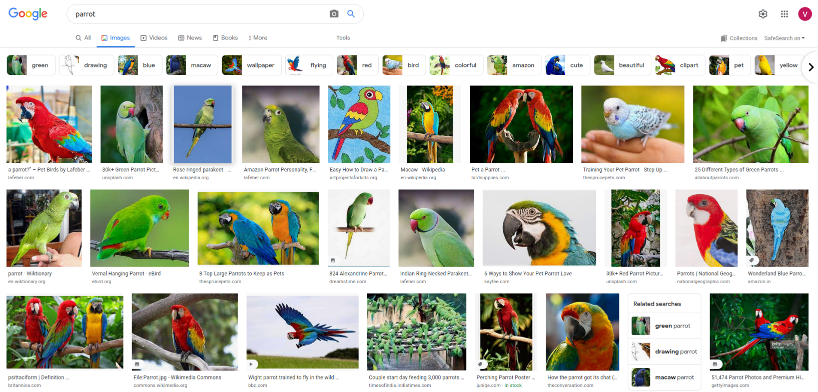



Learning in Vision

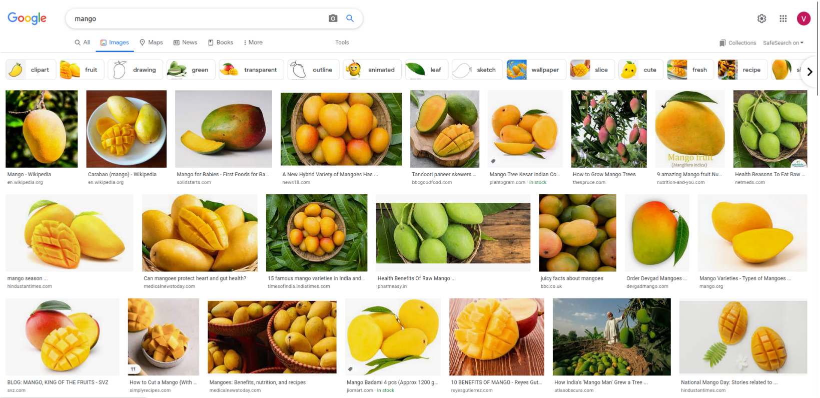


Sources: brittanica.com scmp.com makemytrip.com curlytales.com taj-mahal.net rfi.fr viator.com taj-mahal.net metmuseum.org easydrawing.net

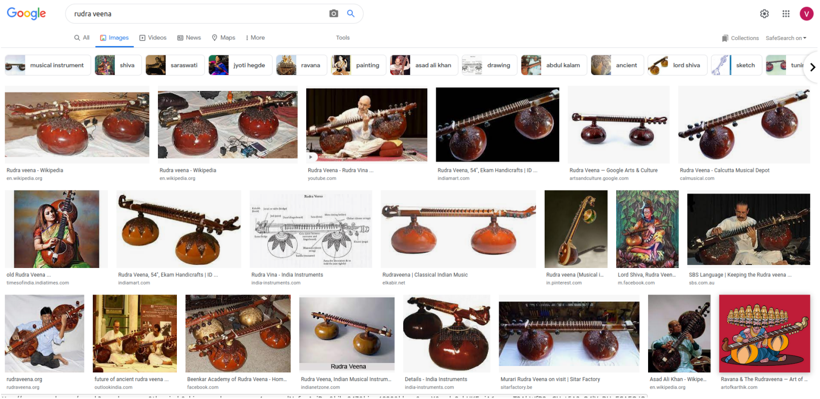
Learning in Vision



Learning in Vision

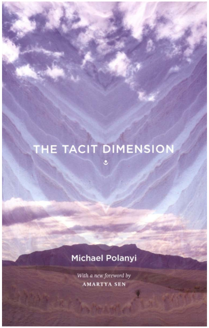


Learning in Vision



Why Learning?

- Higher-level reasoning difficult to model
- Process of reasoning not fully described
- Interested in functional replication
- Flexibility of model
- Biological organisms learn
- Nature vs. Nurture debate



I shall reconsider human knowledge by starting from the fact that we can know more than we can tell. This fact seems obvious enough; but it is not easy to say exactly what it means. Take an example. We know a person's face, and can recognize it among a thousand, indeed among a million. Yet we usually cannot tell how we recognize a face we know. So most of this knowledge cannot be put into words.

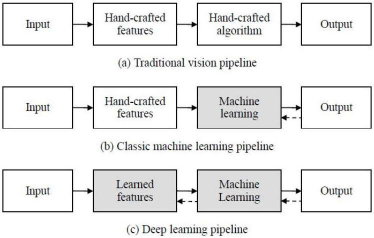
Michael Polanyi
The Tacit Dimension, 1966

Learning in Vision

- Tacit vs Explicit Forms of Knowledge
- Perceptual vs Engineering Solutions
- "All models are wrong, some are useful" to "What models?"
- Polanyi's Revenge

h/t Subbarao Khambampati's talk *Polanyi vs Planning*

Learning in Vision



Why Learning Now?

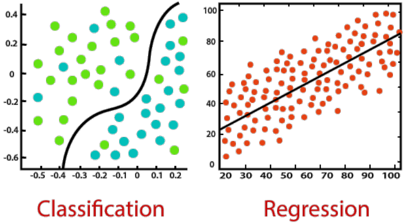
- Low-level vision well developed
- Difficult to formulate general models for reasoning
- Bypass through learning
- Explosion of image data, internet
- Growth of computational power
- **Deep Learning**
- Vision $\neq$ Machine Learning $\neq$ Deep Learning $\neq$ AI!

- Consult slides of Andreas Geiger, Computer Vision (2021) Lecture 10:Recognition
  Link provided on lecture page
  Slide numbers: 3 14-20 60 75 77-82 136 139-140
- Consult slides of Noah Snavely, Introuction to Computer Vision (2021) Lecture 19: Introduction to Recognition
  Link provided on lecture page
  Slide numbers: 12-16 24-29

Topics

- Machine Learning Methods
- Deep Learning and Datasets
- Later: Fairness and Ethics

Learning in Vision

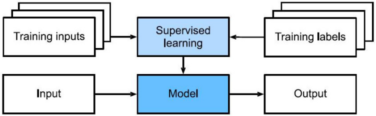


Problems in Learning

- Classification
- Regression
- Clustering

javatpoint.com

Learning in Vision

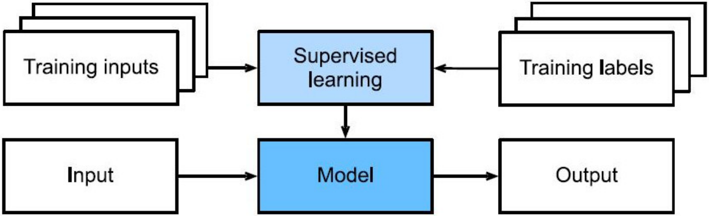


Approaches to Learning

- Supervised
- Unsupervised (self-learning)
- Semi-supervised

Szeliski 2nd Edition

Learning in Vision



Supervised Learning

- Use input-output pairs
- How do we get labels?
- How do we score for tasks?
  - Classification
  - Detection
  - Segmentation

Learning in Vision

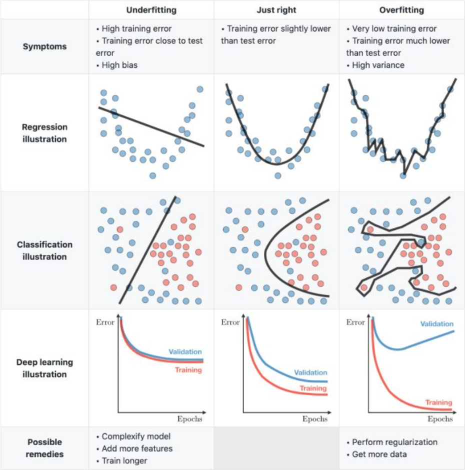
Empirical Risk Minimisation

- $\mathbf{y}_i = f(\boldsymbol{x}_i; \boldsymbol{w})$
- $\Sigma L(\mathbf{y}_i, f(\boldsymbol{x}_i; \boldsymbol{w}))$
- True Risk: $\boldsymbol{E}(L(\boldsymbol{y}, f(\boldsymbol{x}; \boldsymbol{w})))$
- Classification (possibly asymmetric)
- Regression (think line fitting)

Statistical Learning Theory

- This is just a caricature
- Vast body of theoretical work
- Assumption: unknown underlying probability
- Training samples drawn from pdf
- Test from same pdf (Generalisation ?)

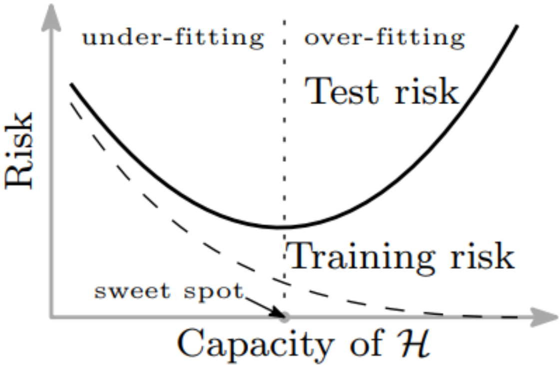
Learning in Vision



https://www.kaggle.com/getting-started/166897

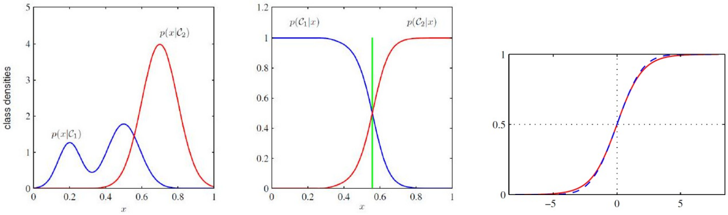
Fitting

- Learning model?
- Expressiveness
- Complexity
- Over vs. underfit
- Deep learning
  - Too many parameters
  - Generalisation?
  - When?



https://blog.ml.cmu.edu/2020/08/31/4-overfitting/

Learning in Vision



Bayes Classifier

$$
\begin{aligned}
p(C_k|\mathbf{x}) &= \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_j p(\mathbf{x}|C_j)p(C_j)} = \frac{\exp l_k}{\sum_j \exp l_j} \\
\text{where } l_k &= \log p(\mathbf{x}|C_k) + \log p(C_k)
\end{aligned}
$$

- Logistic function: $\sigma(l) = \frac{1}{1+e^{-l}}$ for $l = l_0 - l_1$

$$
\begin{aligned}
p(\mathbf{x}|C_k) &= \frac{1}{V_k}\exp\{-\frac{1}{2}(\mathbf{x}-\mu_k)^T\Sigma^{-1}(\mathbf{x}-\mu_k)\} \\
\Rightarrow p(C_0|\mathbf{x}) &= \sigma(\mathbf{w}^T\mathbf{x}+b)
\end{aligned}
$$

Discriminant Analysis

- Binary Classification
- Assume Gaussian distributions (further for 2-class, assume same covariance Σ)
- Result is logistic regression
- Linear Discriminant Function: compare $\mathbf{w}_k^T\mathbf{x} + b_k$
- For non-equal Σ, quadratic discriminant function

Learning in Vision

$$
\begin{aligned}
p_i = p(C_0|\mathbf{x}_i) &= \sigma(\mathbf{w}^T\mathbf{x}_i + b) \\
\Rightarrow E_{CE}(\mathbf{w}, b) &= -\Sigma_i t_i \log p_i + (1 - t_i)\log(1 - p_i)
\end{aligned}
$$

Logistic Regression

- Gaussian assumption too strong
- Work with posterior
- Cross-entropy Loss
- One-hot encoding
- Limitations: when not linearly separable
- Limitations: infinite solutions when separable

Learning in Vision

$$p_{ik} = p(C_k|\mathbf{x}_i) = \frac{\exp l_{ik}}{\Sigma_j \exp l_{ij}} = \frac{1}{Z_i} \exp l_{ik}$$

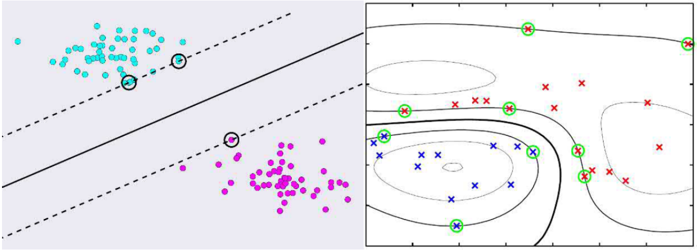
$$\text{with } l_{ik} = \mathbf{w}_k^T \mathbf{x}_i + b_k$$

$$\Rightarrow E_{MCCE}(\mathbf{w}_k, b_k) = -\Sigma_i \Sigma_k \tilde{t}_{ik} \log p_{ik}$$

Logistic Regression

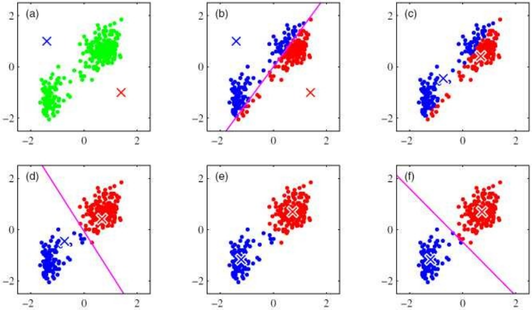
- Gaussian assumption too strong
- Work with posterior
- Cross-entropy Loss
- One-hot encoding
- Limitations: when not linearly separable
- Limitations: infinite solutions when separable

Learning in Vision



Support Vector Machines

- Multiple solutions when separable
- Recognise that data is only partial
- Maximise margin of classifier
- For not linearly separable: kernel regression



Approaches to Learning

- Clustering using k-means

Szeliski 2nd Edition

Learning in Vision

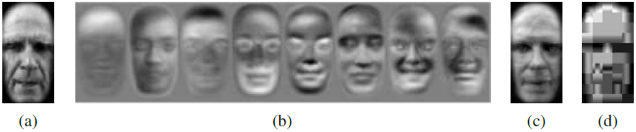
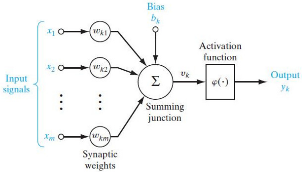


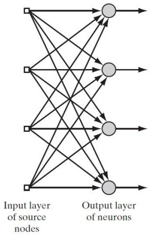
Figure 5.18 *Face modeling and compression using eigenfaces (Moghaddam and Pentland 1997) © 1997 IEEE: (a) input image; (b) the first eight eigenfaces; (c) image reconstructed by projecting onto this basis and compressing the image to 85 bytes; (d) image reconstructed using JPEG (530 bytes).*

Approaches to Learning

- Principal Component Analysis
- $\boldsymbol{C} = \Sigma(\boldsymbol{x}_j - \mu)(\boldsymbol{x}_j - \mu)^T$
- $\boldsymbol{C} = \boldsymbol{U}\Lambda\boldsymbol{U}^T = \sum \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T$
- $\boldsymbol{C} \approx \sum_k \lambda_k \boldsymbol{u}_k \boldsymbol{u}_k^T$
- Low dimensional representation
- Project observation onto subspace

Learning in Vision





Deep Learning

- Simple **nonlinear** model of single neuron
- Old idea of connectionism
- Rosenblatt 1958; Rumelhart *et al.* 1986, Fukushima 1980
- Cycles of interest
- Significant breakthroughs with deep layers
- Dominant paradigm today

Simon Haykin's textbook; Szeliski 2nd Edition

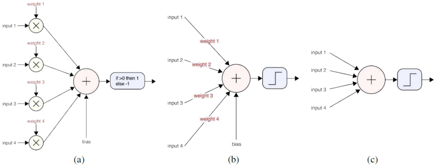
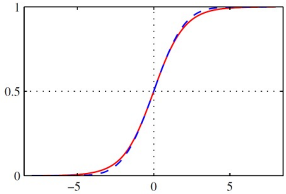


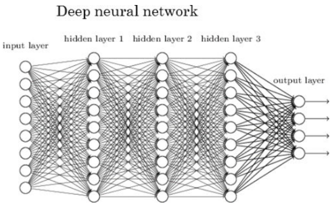
Figure 5.23 A perceptron unit (a) explicitly showing the weights being multiplied by the inputs, (b) with the weights written on the input connections, and (c) the most common form, with the weights and bias omitted. A non-linear activation function follows the weighted summation. © Glassner (2018)

Perceptron Model

- Feedforward networks
- Simple "neurons", rich connections
- $y = h(s) = h(\boldsymbol{w}^T \boldsymbol{x} + b)$
- $h(l) = \frac{1}{1+e^{-l}}$
- Key: Non-linearity of neuron



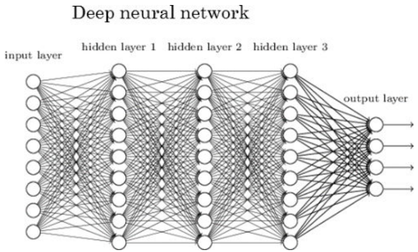
Simon Haykin's textbook; Szeliski 2nd Edition



Multilayer Neural Networks

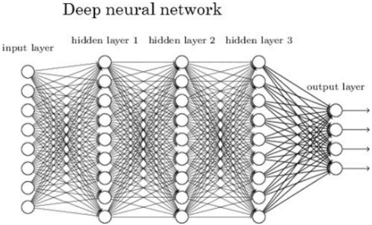
- Regular structure with layers
- Each layer outputs: $\mathbf{s}_l = \mathbf{W}_l \mathbf{x}_l$
- Next layer: $\mathbf{x}_{l+1} = \mathbf{y}_l = h(\mathbf{s}_l)$
- Output: $\mathbf{y} = h_{\mathbf{W}_N}(h_{\mathbf{W}_{N-1}}((\cdots(\boldsymbol{x}))))$
- Non-linear function mapping: $\mathbf{y} = H(\mathbf{x}, \mathbb{W})$
- $\mathbb{W}$: All weights in all layers!
- Expressive power

Learning in Vision



Deep Neural Networks

- What is deep here?
- Non-linear with **many many** weights!
- Breakthrough in 2012
- Tsunami of DL approaches
- Completely taken over vision and ML (almost)

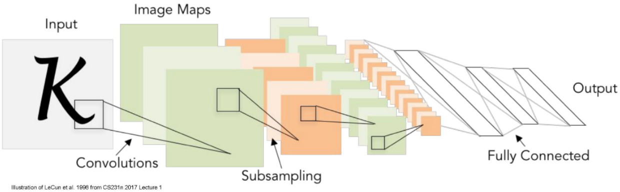


Types of Neural Networks

- Layers with vector inputs
- Convolutional Networks (Receptive Fields)
- Temporal Networks (LSTM, Transformer)
- Many more models

Noah Snavely's slides; Kevin Murphy's book

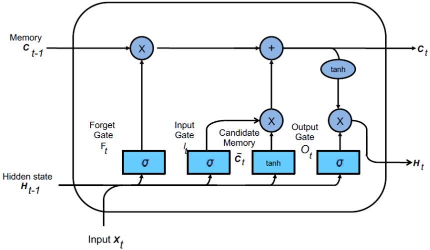
Learning in Vision



Types of Neural Networks

- Layers with vector inputs
- Convolutional Networks (Receptive Fields)
- Temporal Networks (LSTM, Transformer)
- Many more models

Noah Snavely's slides; Kevin Murphy's book

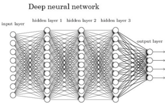


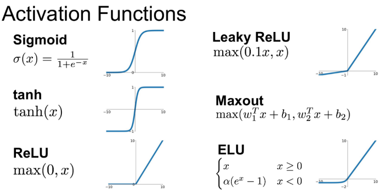
Types of Neural Networks

- Layers with vector inputs
- Convolutional Networks (Receptive Fields)
- Temporal Networks (LSTM, Transformer)
- Many more models

Noah Snavely's slides; Kevin Murphy's book

Learning in Vision



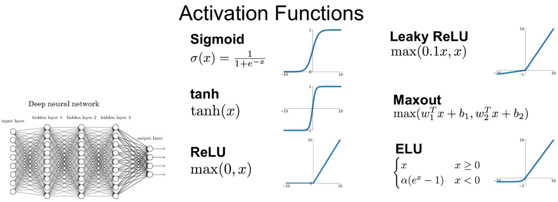


Key Ingredients

- Non-linear activation functions
- Gradient descent for fitting
- Learning over masses of data
- Nested functions $h(h(h(\cdots)))$
- Derivatives using chain rule of calculus
- Learning through **Backpropagation**
- Stochastic Gradient Descent

https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092

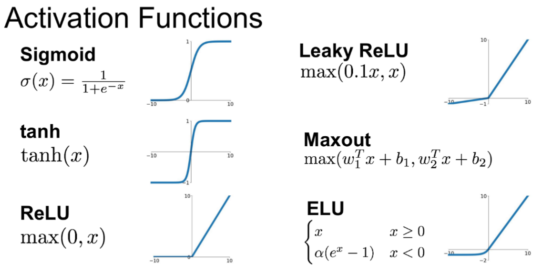
Learning in Vision



Key Ingredients

- Non-linear activation functions
- Gradient descent for fitting
- Learning over masses of data
- Nested functions $h(h(h(\cdots)))$
- Derivatives using chain rule of calculus
- Learning through **Backpropagation**
- Stochastic Gradient Descent (Graduate Student Descent)

https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092



Activation Functions

- Many functions
- Sigmoid is smooth
- ReLU is simple and popular
- ReLU has issues

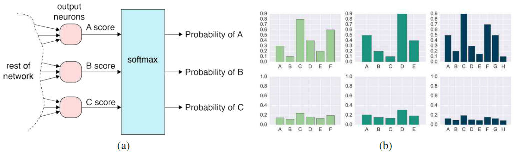


Figure 5.27 (a) A softmax layer used to convert from neural network activations ("score") to class likelihoods (b) The top row shows the activations, while the bottom shows the result of running the scores through softmax to obtain properly normalized likelihoods. © Glassner (2018).

Softmax Layer

- $p_i = \frac{\exp x_i}{\sum_k \exp x_k}$
- *Soft* version of *max*
- Often as last layer
- Converts outputs to class likelihoods

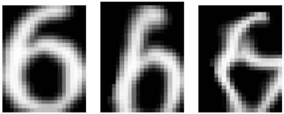
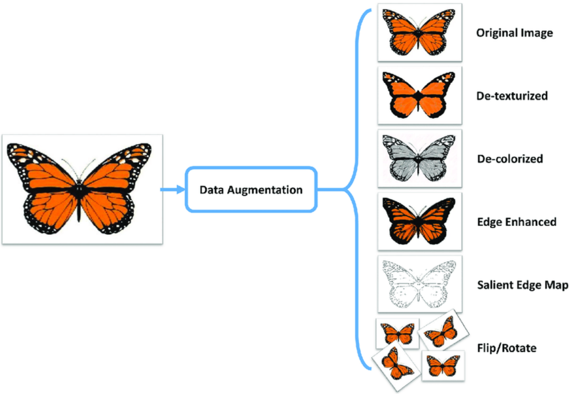


Figure 5.28 An original "6" digit from the MNIST database and two elastically distorted versions (Simard, Steinkraus, and Platt 2003) © 2003 IEEE.

Data Augmentation

- Use training samples
- Reduce over-fitting
- Augment traning data with distortions

Learning in Vision

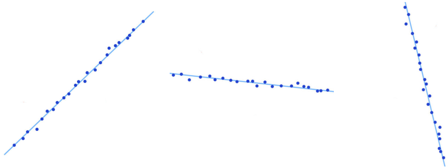


Data Augmentation

- Variety of augmentations in range and domain
- Very hacky

https://medium.com/@sauravkumarsct

Learning in Vision



Invariances and Equivariances

- Invariance: Output doesn't change with nuisance variable
- Equivariance: Invariance upto equivariant factor
- $\mathbf{l}^T\mathbf{p} = (\mathbf{R}\mathbf{l})^T(\mathbf{R}\mathbf{p}) = 0$
- Line fitting using different co-ordinate systems
- Recall OLS vs. TLS solutions
- Deep Learning can fail catastrophically
- Recent approaches more principled

Learning in Vision

Figure 5.50 *Examples of adversarial images from © Szegedy, Zaremba* et al. *(2013). For each original image in the left column, a small random perturbation (shown magnified by 10× in the middle column) is added to obtain the image in the right column, which is always classified as an ostrich.*

Learning can be Brittle

- Catastrophic failures
- Why does this happen?
- Explainable approaches
- GANs

Learning in Vision

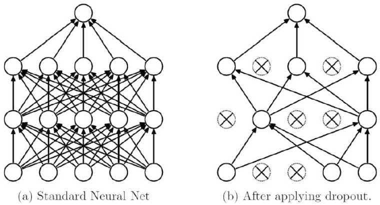


Figure 5.29 When using *dropout*, during training some fraction of units p is removed from the network (or, equivalently, clamped to zero) © Srivastava, Hinton *et al.* (2014). Doing this randomly for each mini-batch injects noise into the training process (at all levels of the network) and prevents the network from overly relying on particular units.

Dropout

- Method for regularization
- Reduces overfitting, improves generalization
- Applies to each mini-batch

Szeliski 2nd Edition

Learning in Vision

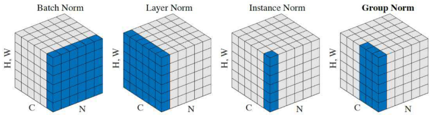


Figure 5.30 Batch norm, layer norm, instance norm, and group norm, from Wu and He (2018) © 2018 Springer. The (H, W) dimension denotes pixels, C denotes channels, and N denotes training samples in a minibatch. The pixels in blue are normalized by the same mean and variance.

Batch Normalization

- Optimization is tricky, needs good conditions
- Recall condition number, scaling
- Varying scales of weights, outputs
- Components of gradient scaled differently
- Simple scaling+recentering along layers etc.

Loss Functions

- Define optimization cost or loss
- Classification vs. Regression
- Classification: Cross-entropy loss
- Contrastive learning, metric embedding
- Regression: typically least squares
- Issue: our confidence in output

Szeliski 2nd Edition

Learning in Vision

Supervised

Given "ground truth" F for training data

$\min_{\mathbb{W}} \sum_k ||H(\boldsymbol{X}^k, \mathbb{W}) - F_k||^2$

Unsupervised

$\min_{\mathbb{W}} \sum_k \sum_i \rho(\boldsymbol{x}_i^{\prime kT} H(\boldsymbol{X}^k, \mathbb{W}) \boldsymbol{x}_i^k)^2$

Learning Epipolar Geometry

- Toy example illustration
- Supervised vs. Unsupervised Learning
- Correspondences $\boldsymbol{X} = \{(\boldsymbol{x}_i, \boldsymbol{x}_i^{\prime}) | i = 1, \cdots, N\}$
- Can contain outliers
- Learnt model $F = H(\boldsymbol{X}, \mathbb{W})$
- Recall IRLS weights $\boldsymbol{W} = \{w_i, i = 1 \cdots N\}$
- Learn to estimate weights directly $\boldsymbol{W}_{correspondences} = H(\boldsymbol{X}, \mathbb{W})$
- $\mathbf{W}_{correspondences}$: Not to be confused with network weights $\mathbb{W}$!
- "Learning to Find Good Correspondences"

$$\min_{\mathbb{W}} \sum_{k} ||\mathbf{y} - H(\mathbf{x}_k, \mathbb{W})||^2$$

Solving for Weights

- Learning is an optimization problem
- Optimize what?
- How?
- Too much data for higher-order methods
- Key observation: two passes

$$\min_{\mathbb{W}} \sum_k ||\mathbf{y} - H(\mathbf{x}_k, \mathbb{W})||^2$$

Solving for Weights

- Learning is an optimization problem
- Optimize what? Weights $\mathbb{W}$
- How?
- Too much data for higher-order methods
- Key observation: two passes

Learning in Vision

$$\min_{\mathbb{W}} \sum_k ||\mathbf{y} - H(\mathbf{x}_k, \mathbb{W})||^2$$

Solving for Weights

- Learning is an optimization problem
- Optimize what? Weights $\mathbb{W}$
- How?
- Too much data for higher-order methods
- Key observation: two passes

Learning in Vision

$$\min_{\mathbb{W}} \sum_k ||\mathbf{y} - H(\mathbf{x}_k, \mathbb{W})||^2$$

Solving for Weights

- Learning is an optimization problem
- Optimize what? Weights $\mathbb{W}$
- How? Gradient Descent
- Too much data for higher-order methods
- Key observation: two passes

$$\min_{\mathbb{W}} \sum_k ||\mathbf{y} - H(\mathbf{x}_k, \mathbb{W})||^2$$

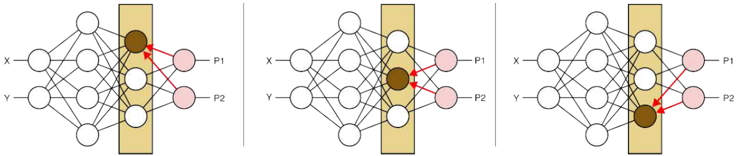


Figure 5.31 Backpropagating the derivatives (errors) through an intermediate layer of the deep network © Glassner (2018). The derivatives of the loss function applied to a single training example with respect to each of the pink unit inputs are summed together and the process is repeated chaining backward through the network.

Solving for Weights

- Learning is an optimization problem
- Optimize what? Weights $\mathbb{W}$
- How? Gradient Descent
- Too much data for higher-order methods
- Key observation: two passes

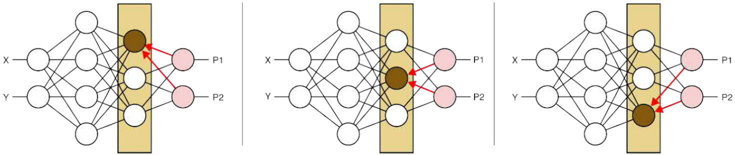


Figure 5.31 Backpropagating the derivatives (errors) through an intermediate layer of the deep network © Glassner (2018). The derivatives of the loss function applied to a single training example with respect to each of the pink unit inputs are summed together and the process is repeated chaining backward through the network.

Backpropagation

- Backpropagation: Rumelhart, Hinton, Williams (1986)
- Compute output in *forward* pass
- Want to change weights $\mathbb{W}$ in descent direction
- Derivative of output wrt input $\mathbf{x}_k$?
- Summation of individual contributions
- Derivative of output wrt weights?

Learning in Vision

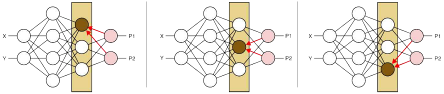


Figure 5.31 Backpropagating the derivatives (errors) through an intermediate layer of the deep network © Glassner (2018). The derivatives of the loss function applied to a single training example with respect to each of the pink unit inputs are summed together and the process is repeated chaining backward through the network.

Backpropagation

- Recall $y = H(\mathbf{x}, \mathbb{W}) = h_{\mathbf{W}_N}(h_{\mathbf{W}_{N-1}}((\cdots(x))))$
- Loss: $E = (y - H(\mathbf{x}, \mathbb{W}))^2$
- Denote $y_i = h(s_i) = h(\mathbf{w}_i^T \mathbf{x})$
- $\frac{\partial E}{\partial s_i} = h'(s_i) \frac{\partial E}{\partial y_i}$
- What does y_i depend on?
- $y = h(h(h(\cdots)))$

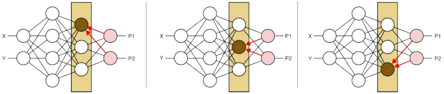


Figure 5.31 Backpropagating the derivatives (errors) through an intermediate layer of the deep network © Glassner (2018). The derivatives of the loss function applied to a single training example with respect to each of the pink unit inputs are summed together and the process is repeated chaining backward through the network.

Backpropagation

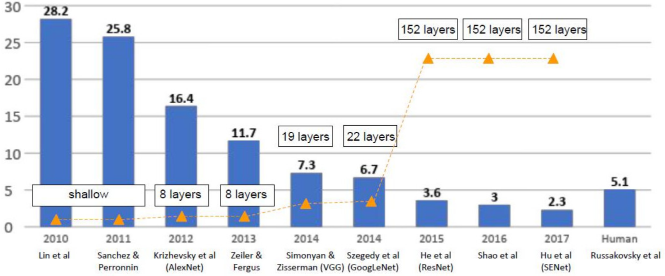
- Recall y_i depends on outputs of previous layer
- Recall y_i affects subsequent layers
- Define ‘error’ $e_i = \frac{\partial E}{\partial s_i}$
- $\frac{\partial E}{\partial y_i} = \sum_{k>i} \frac{\partial E}{\partial x_{ki}} = \sum_{k>i} w_{ki} e_k$
- $e_i = h^{'}(s_i) \frac{\partial E}{\partial y_i} = h^{'}(s_i) \sum_{k>i} w_{ki} e_k$
- Chain rule: Derivative of loss (error) wrt unit
- Depends on weighted sum of errors of units feeds into
- Store activations in forward pass
- Estimate in backward sweep (bfs)

$$
\begin{aligned}
\mathbf{w}_{t+1} &= \mathbf{w}_t - \alpha \mathbf{g} \\
\text{Define } \mathbf{v}_{t+1} &= \rho \mathbf{v}_t + \mathbf{g}_t \\
\mathbf{w}_{t+1} &= \mathbf{w}_t - \alpha \mathbf{v}_t \text{ with momentum}
\end{aligned}
$$

Training Issues

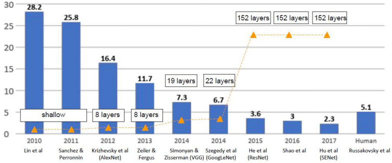
- Data too big for higher-order methods
- Just use gradient descent
- Gradient: sum of gradient terms of each $\boldsymbol{x}$
- Stochastic Gradient Descent
- Minibatches: $\cdots[\cdots][\cdots][\cdots]\cdots$
- Epoch: One cycle through batches
- α: learning rate to be annealed (why?)
- ρ is relatively large
- Hyper-parameters

Learning in Vision



Key Ingredients

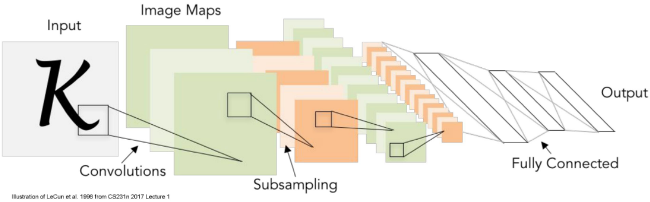
- Large datasets are important
- Deep Networks
- Massive Compute Power
- AlexNet: 8 Layers; ResNet: 152 layers
- ImageNet Dataset: 1000 classes, $>$ million images



Ethics of datasets

- Transparency of acquisition process, privacy
- Ethics problems should not be ignored
- Large % of images removed from ImageNet
- Ethics of labour (Amazon Mechanical Turk)
- Obsession with test error
- “Datasheets for Datasets”

Szeliski 2nd Edition

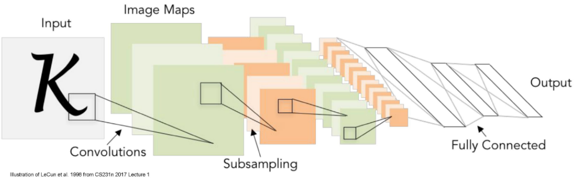


Deep Learning for Images

- Convolutional Neural Networks
- Locality of pixels propagated
- End-to-end learning
- Unified approaches for multiple tasks
- Segmentation, Localization, Recognition

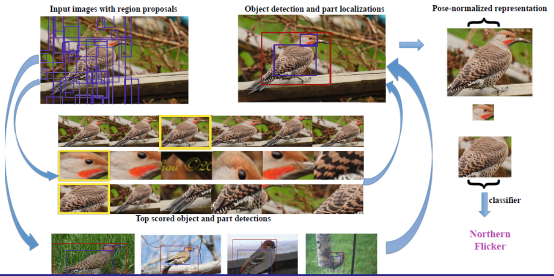
Kevin Murphy's book

- Consult slides of Noah Snavely, Introduction to Computer Vision (2021)
  Lecture 21: Convolutional Neural Networks
  Link provided on lecture page
  Slide numbers: 57-100



Object Recognition

- Major breakthroughs in recognition tasks
- Efficient computation of repeated convolutions
- Older approaches: Instance Recognition
  - re-recognise specific objects
- Current approaches: Class or Category Recognition
  - Variable classes: dogs, cats, chairs
- Fine-grained categories



Object Recognition

- Major breakthroughs in recognition tasks
- Efficient computation of repeated convolutions
- Older approaches: Instance Recognition
  - re-recognise specific objects
- Current approaches: Class or Category Recognition
  - Variable classes: dogs, cats, chairs
- Fine-grained categories

Learning in Vision

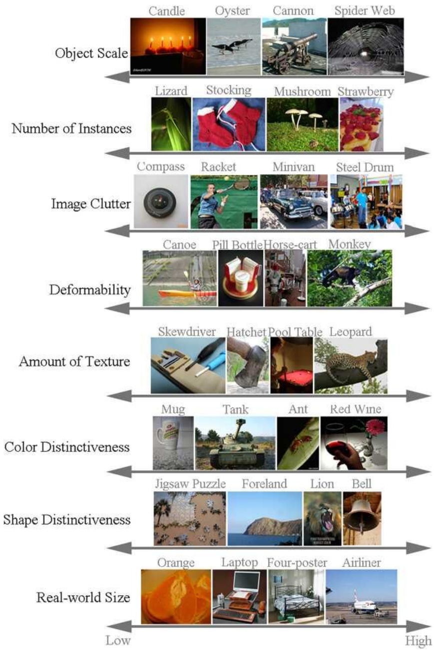


Image Net Examples; Szeliski 2nd edition

Learning in Vision

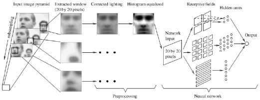


Figure 6.21 *A neural network for face detection (Rowley, Baluja, and Kanade 1998) © 1998 IEEE. Overlapping patches are extracted from different levels of a pyramid and then pre-processed. A three-layer neural network is then used to detect likely face locations.*

Object Detection

- Early work in detecting faces, people (pedestrians)
- Early neural networks
- Some used bag of words
- Deformable parts model
- Boosting: Combine many simple features
- Cascade of classifiers

Szeliski 2nd edition

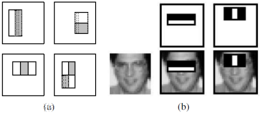


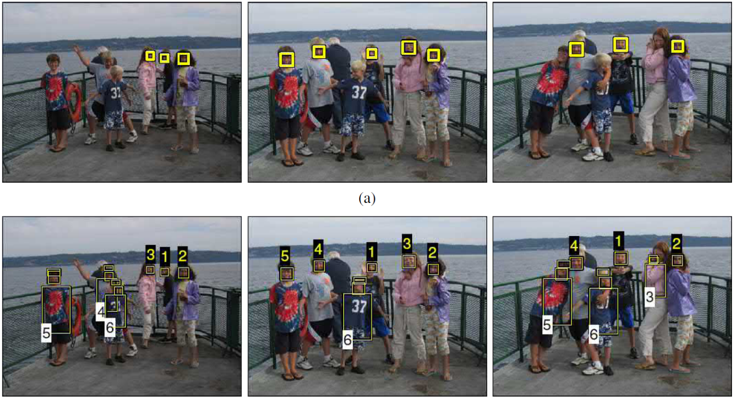
Figure 6.22 *Simple features used in boosting-based face detector (Viola and Jones 2004) © 2004 Springer: (a) difference of rectangle feature composed of 2–4 different rectangles (pixels inside the white rectangles are subtracted from the gray ones); (b) the first and second features selected by AdaBoost. The first feature measures the differences in intensity between the eyes and the cheeks, the second one between the eyes and the bridge of the nose.*

Object Detection

- Early work in detecting faces, people (pedestrians)
- Early neural networks
- Some used bag of words
- Deformable parts model
- Boosting: Combine many simple features
- Cascade of classifiers

Szeliski 2nd edition

Learning in Vision



(a)

(b)

Figure 6.19 Person detection and re-recognition using a combined face, hair, and torso model (Sivic, Zitnick, and Szeliski 2006) © 2006 Springer. (a) Using face detection alone, several of the heads are missed. (b) The combined face and clothing model successfully re-finds all the people.

Learning in Vision

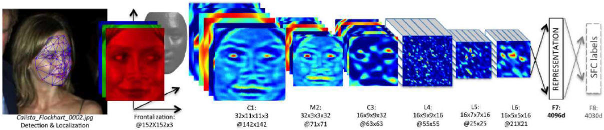


Figure 6.17 *The DeepFace architecture (Taigman, Yang* et al. *2014) © 2014 IEEE, starts with a frontalization stage, followed by several locally-connected (non-convolutional) layers, and then two fully connected layers with a K-class softmax.*

Face Recognition

- High interest: Access, surveillance
- Seen PCA version earlier (EigenFaces)
- DL version: Frontalization + Recognition
- Works well in many contexts
- Accuracy "in the wild" is questionable
- **Extraordinary** crises around FRT
- Discuss in Ethics lecture

Learning in Vision

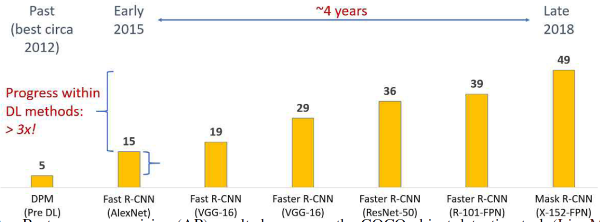
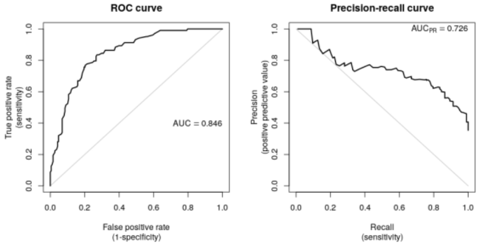


Figure 6.29 Best average precision (AP) results by year on the COCO object detection task (Lin, Maire*etal*. 2014) © 2020 Ross Girshick.

Generic Object Detection

- Major breakthroughs with DL
- Rectangular regions
- Based on sliding window tests



How to Score Performance?

- Two types of errors
- Receiver Operating Characteristic (ROC)
    - True Positive vs. False Positive
- Precision-Recall (PC)
    - True, False, Number of Positives (TP,FP,NP)
    - Precision=$\frac{TP}{TP+FP}$
    - Recall=$\frac{TP}{NP}$
- Average Precision (AP);*mean*AP (mAP) over all categories

https://www.r-bloggers.com/2020/01/area-under-the-precision-recall-curve/

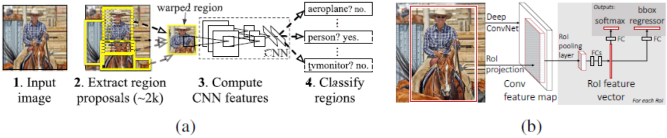


Figure 6.28 *The R-CNN and Fast R-CNN object detectors. (a) R-CNN rescales pixels inside each proposal region and performs a CNN + SVM classification (Girshick, Donahue* et al. *2015) © 2015 IEEE. (b) Fast R-CNN resamples convolutional features and uses fully connected layers to perform classification and bounding box regression (Girshick 2015) © 2015 IEEE.*

Modern Object Detectors

- Rectangular Region Proposals + Classifier
- R-CNN: Region-based CNN
  - $\approx$ 2000 region proposals
  - Each warped to fixed 224×224 region
  - Classify using SVM

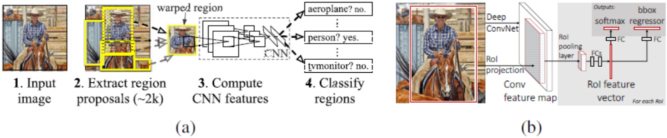


Figure 6.28 *The R-CNN and Fast R-CNN object detectors. (a) R-CNN rescales pixels inside each proposal region and performs a CNN + SVM classification (Girshick, Donahue* et al. *2015) © 2015 IEEE. (b) Fast R-CNN resamples convolutional features and uses fully connected layers to perform classification and bounding box regression (Girshick 2015) © 2015 IEEE.*

Modern Object Detectors

- Rectangular Region Proposals + Classifier
- Fast R-CNN
  - End-to-end
  - Resamples convolution features for proposals
  - Classify using fully connected network

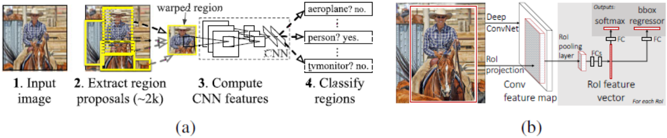


Figure 6.28 *The R-CNN and Fast R-CNN object detectors. (a) R-CNN rescales pixels inside each proposal region and performs a CNN + SVM classification (Girshick, Donahue* et al. *2015) © 2015 IEEE. (b) Fast R-CNN resamples convolutional features and uses fully connected layers to perform classification and bounding box regression (Girshick 2015) © 2015 IEEE.*

Modern Object Detectors

- Rectangular Region Proposals + Classifier
- Also Faster R-CNN
- Single network for detection+classification
  - Single Shot Multibox Detector (SSD)
  - You Only Look Once (YOLO)

Learning in Vision

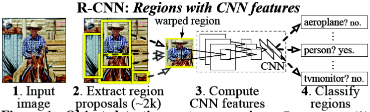


Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7% on PASCAL VOC 2010**. For comparison, [39] reports 35.1% mAP using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. The popular deformable part models perform at 33.4%. On the 200-class **ILSVRC2013 detection dataset, R-CNN's mAP is 31.4%**, a large improvement over OverFeat [34], which had the previous best result at 24.3%.

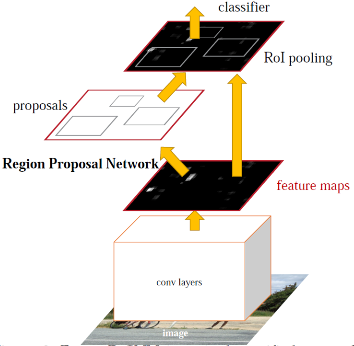
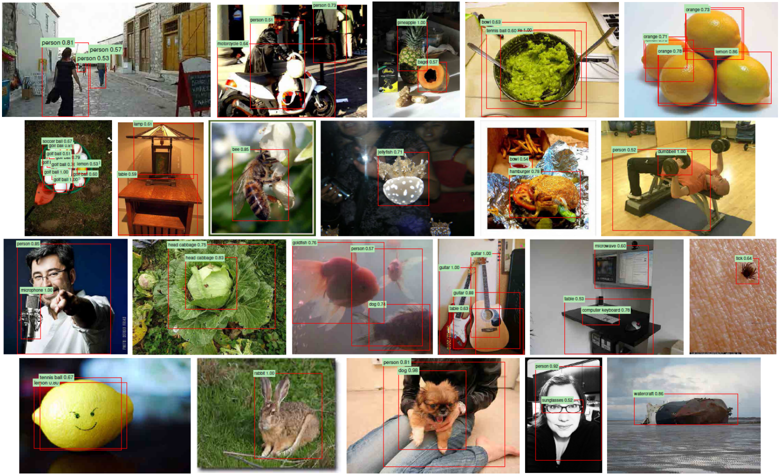


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

RCNN and Faster RCNN papers



Some results from RCNN paper

Learning in Vision

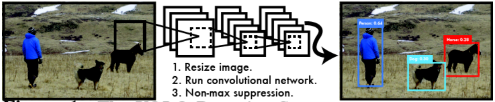
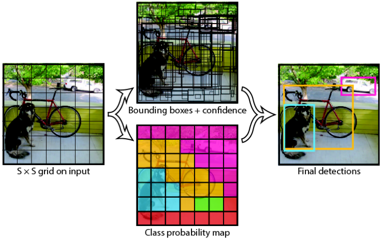


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

You Only Look Once

- Single shot instead of two-stages
- Directly predicts 2D bounding box
- Faster, lower performance
- Redmon *et al.*, 'You Only Look Once: Unified, Real-Time Object Detection', CVPR 2016
- Many improvements
- Ethics dimensions in next lecture



You Only Look Once

- Single shot instead of two-stages
- Directly predicts 2D bounding box
- Faster, lower performance
- Redmon *et al.*, 'You Only Look Once: Unified, Real-Time Object Detection', CVPR 2016
- Many improvements
- Ethics dimensions in next lecture

Learning in Vision

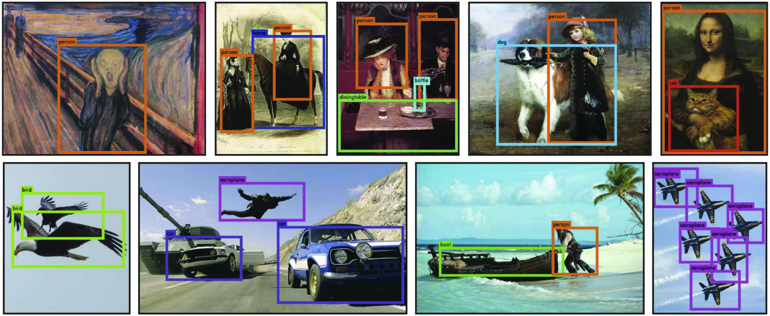
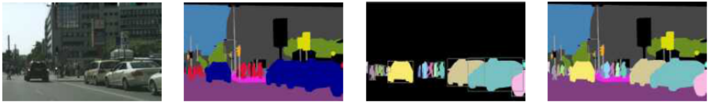


Figure 6: Qualitative Results. YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane.

Results from YOLO paper

Learning in Vision



(a) (b) (c) (d)

Figure 6.32 *Examples of image segmentation (Kirillov, He* et al. *2019) © 2019 IEEE: (a) original image; (b) semantic segmentation (per-pixel classification); (c) instance segmentation (delineate each object); (d) panoptic segmentation (label all things and stuff).*

Semantic Segmentation

- Standard segmentation: distinction between classes
- Pairwise potentials: similarity + proximity
- No classification
- Semantic segmentation: per-pixel classification
- Networks "percolate" semantic information to pixels

(a) (b)

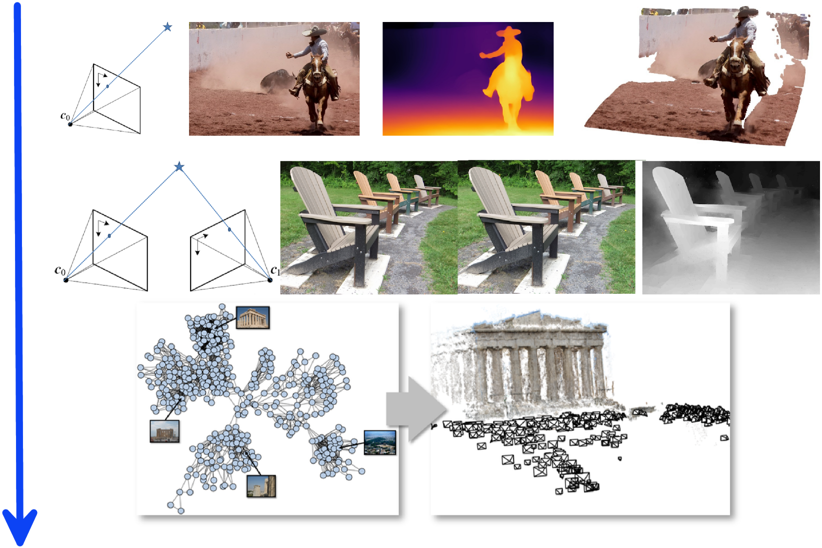
Figure 6.36 *Instance segmentation using Mask R-CNN (He, Gkioxari et al. 2017) © 2017 IEEE: (a) system architecture, with an additional segmentation branch; (b) sample results.*

Instance Segmentation

- Find all objects, give per-pixel masks
- Mask R-CNN
  - Region proposal as Faster R-CNN
  - Additional branch for mask prediction
  - Training loss carefully combines all parts

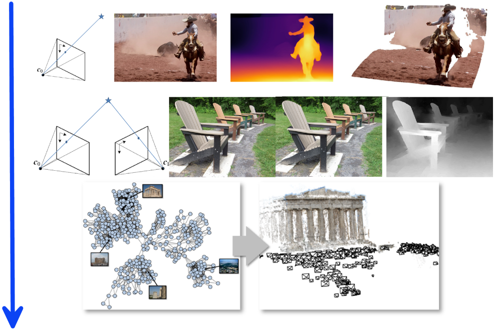
- Consult slides of Andreas Geiger, Computer Vision (2021) Lecture 9: Co-ordinate Based Networks
  Link provided on lecture page
  Slide numbers: 54-66

Learning in 3D Geometry Estimation



Progression from Tacit to Explicit Problems

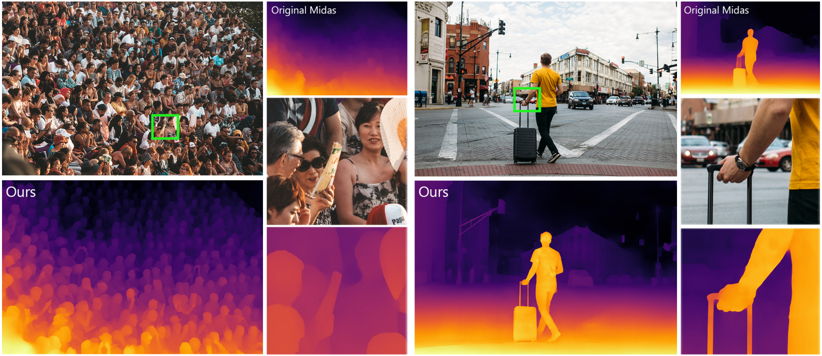
Ranftl *et al.*, ‘Towards Monocular Depth Estimation’;http://3dstereophoto.blogspot.com; https://www.cs.cornell.edu/projects/bigsfm



3D Geometry Problems

- Correspondence is ambiguous for low texture
- ∴ **dense depth** estimation has tacit parts
- Geometric problems with explicit forms
  - camera motion estimation
  - sparse triangulation for corners
- Recognise distinction between tacit and explicit aspects
- Implications for accuracy and reliability

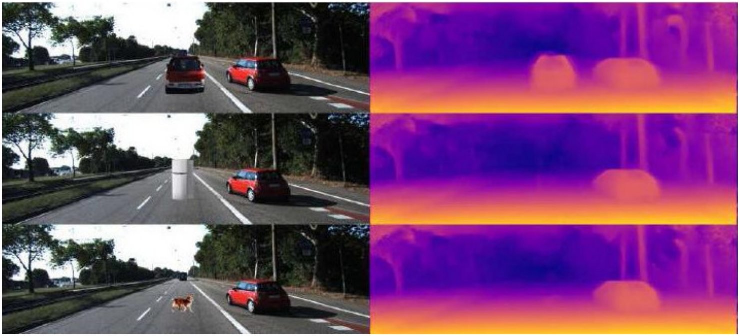
Ranftl *et al.*, 'Towards Monocular Depth Estimation';http://3dstereophoto.blogspot.com; https://www.cs.cornell.edu/projects/bigsfm



Monocular Depth

- Very impressive, but what kind of depth is it?
- Notions of depth: Euclidean, quasi-Euclidean, ordinal, bounding box
- Semantic segmentation of depth is useful for tasks

Miangoleh *et al.*, 'Boosting Monocular Depth ...', CVPR 2021



Monocular Depth

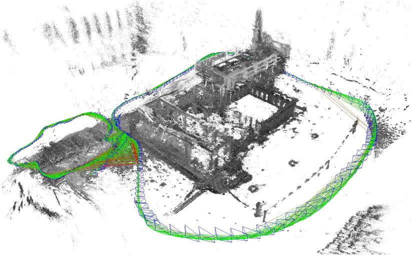
- Learnt models for specific narrow contexts
- Lessons
  - networks ignore apparent size
  - use vertical position of objects
  - dark region used to detect obstacles
  - brittle and unreliable

van Dijk *et al.*, 'How Do Neural Networks See Depth in Single Images?', ICCV 2019

Two-View Stereo

- Recover dense depth with known geometry
- Stereo is a correspondence problem
- Many ambiguities and issues
- Search constraint + ambiguous correspondence
- $\Rightarrow$ mixture of explicit and tacit problems

http://3dstereophoto.blogspot.com

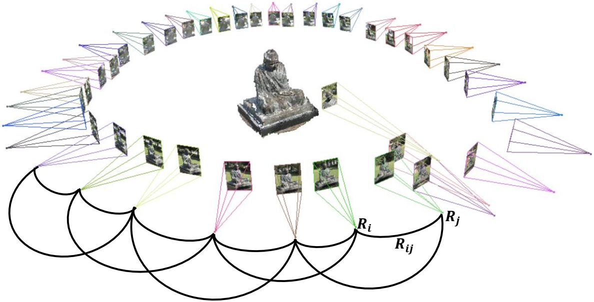


Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.

Dense models of several landmarks produced by COLMAP's MVS pipeline.

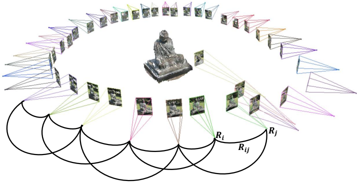
3D Reconstruction from Many Images

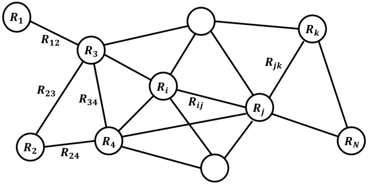
- Geometry induced by pinhole camera
- SLAM vs SfM
- Significantly different motion and noise distributions
- Implications for use of **brightness constraint**



Global Approaches to SfM

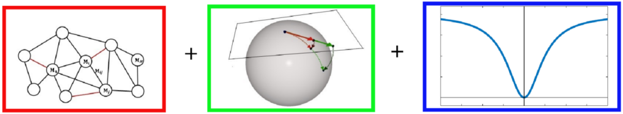
- Jointly solve geometry over all cameras
- Many two-view relative motions available
- Averaging: Solve global **rotations** and translations
- Solve for 3D structure and refine





Rotation Averaging

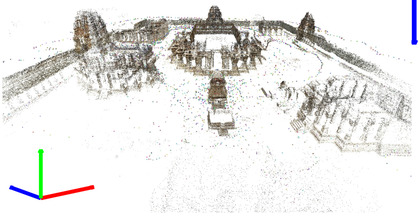
- Viewgraph of camera-camera relations
- Given $\mathbf{R}_{ij}$ on each edge
- Solve for individual cameras $\boldsymbol{R}_i$
- Use relationship: $\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^{-1}$
- Optimisation of robust geometric cost

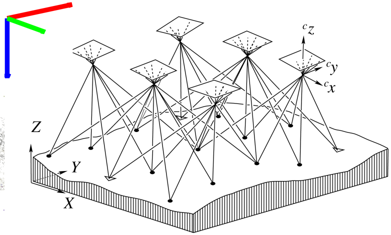


$$\sum_{\mathcal{E}} \rho\left(d\left(\boldsymbol{R}_{ij}, \boldsymbol{R}_j \boldsymbol{R}_i^{-1}\right)\right)$$

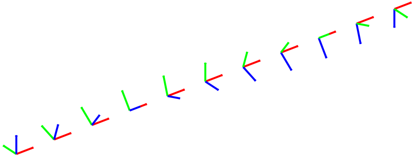
Rotation Averaging

- Deep learning does well compared to geometric methods
- Key factors
  - **Distribution of rotations**
  - **Distribution of noise+outliers**
  - **Distribution of viewgraph edges**
- Combinatorial explosion
- **Is accuracy on datasets enough?**
- **What is learnt?**
- **How reliable are learnt models?**



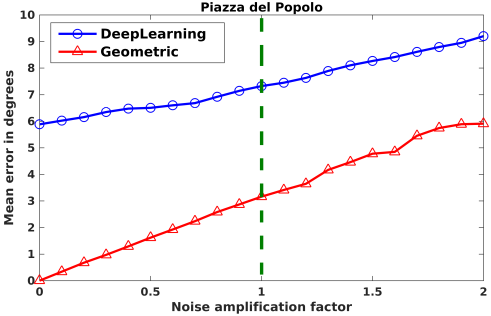


Forstner, *Photogrammetric Computer Vision*



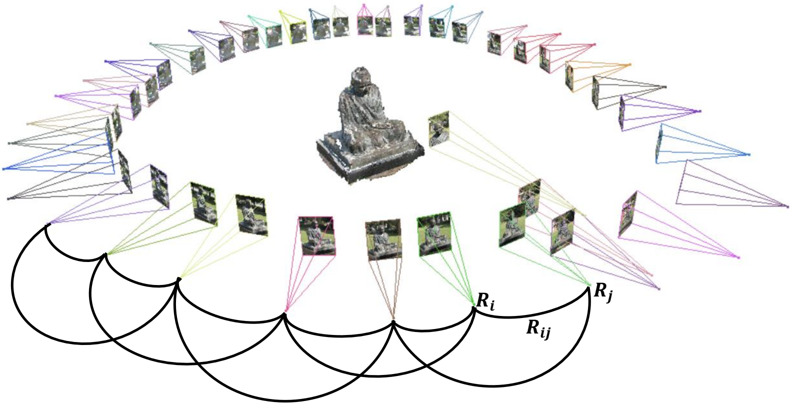
Gauge Freedom

- Arbitrary choice of basis
- Rotations should be equivariant
- Natural for geometric methods
- Not for learnt models



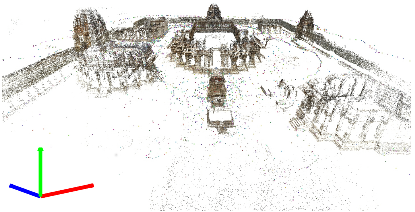
Robustness

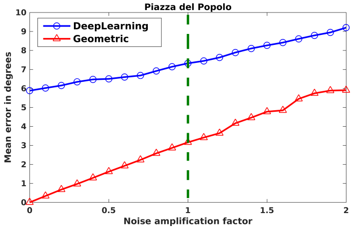
- Good performance on noisy real-world SfM datasets
- Consider perfect data: $\boldsymbol{R}_{ij} = \boldsymbol{R}_j \boldsymbol{R}_i^{-1}$ exactly
- Exact solution exists
- DL method has non-zero error
- What has it learnt?

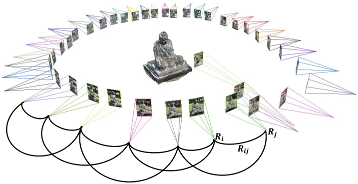


SLAM sequences

- Smooth sequences
  - dense connectivity
  - small rotations
- Loop closures are very useful
- DL method trained on SfM data fails here







	Geometric Method	Deep Learning
Equivariance	✓	✗
Robustness	✓	?
Graph Agnostic	✓	✗
Loop Closure	✓	✗

Some Observations

- Geometry is fundamental in vision
- Desired accuracy: qualitative vs. metric
- Limitations are understood: ambiguous configurations, high noise, outliers
- Deep Learning for geometry
  - works well in narrow contexts
  - combinatorial explosion difficult to tame
  - lacks desirable properties
  - can be unreliable

Some Observations

- DL to mitigate geometric ambiguities + limitations
- Useful for
  - tacit parts of 3D reconstruction pipeline
  - weights for robust least squares
  - initialisation of geometry
  - principled fusion with geometric estimates

Summary

- *Almost* all problems now have DL version
- Datasets play key role in developments
- More (layers) the merrier?
- Massive computational power involved
- Vision tools with high accuracies (deployable)
- What does such “learning” mean?
- Debates on AGI
- Pitfalls: Safety, Privacy, Accuracy, Ethics
- Data+Computational Divide between haves and have-nots
- Handful of corporations driving agenda
- Environmental impact of deep learning
- Deep Learning will continue to dominate
- Consequences?