Soliton

Vision for a Better World

Deep Learning in Computer Vision

Anthill 2018

The team

Sumod

Founder, AutoInfer
&
CV & ML Architect,
Soliton Technologies

Shivaraj

3D Vision Lead,
Computer Vision and
Machine Learning,
Soliton Technologies

Dhivakar

Senior R&D Engineer,
Computer Vision and
Machine Learning,
Soliton Technologies

Senthil

R&D Engineer,
Computer Vision and
Machine Learning,
Soliton Technologies

Soliton
Vision for a Better World

3D Computer Vision

- Camera Calibration - slides from day1
- Human Eye
- Stereo Camera Setup
- Epipolar Geometry
- Essential Matrix and Fundamental Matrix
- Depth Map Calculation from Stereo Images
- Essential Matrix Decomposition
- Triangulation from Two Views
- Triangulation from Multiple Views & Bundle Adjustment
- 3D Reconstruction Steps
- SLAM Introduction
- Demo of SLAM

Camera Model

Lens configuration (internal parameter)

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = L\left(\mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \right)$$

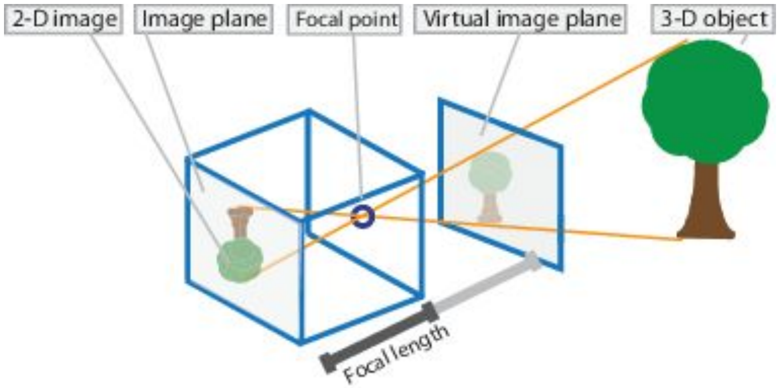
Spatial relationship between sensor and pinhole (internal parameter)

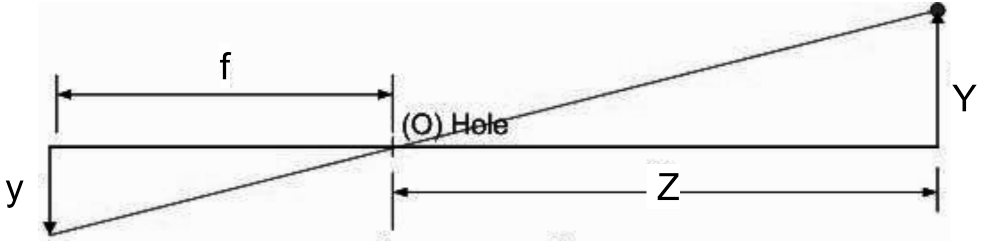
Camera body configuration (extrinsic parameter)

Pinhole Camera Model - University of Pennsylvania | Coursera

Pinhole Camera Model

- Simplest model of imaging process





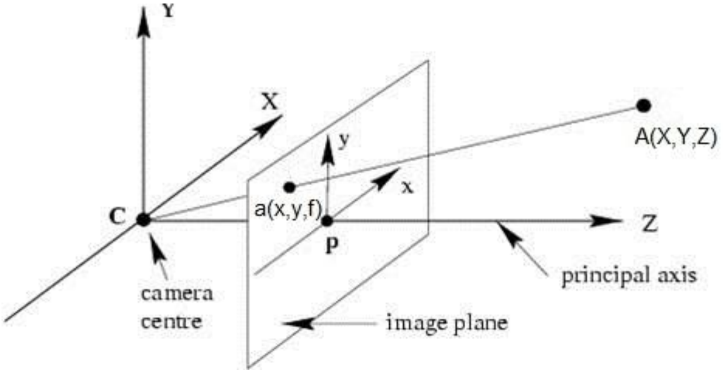
$$\frac{y}{f} = \frac{Y}{Z} \longrightarrow y = \frac{fY}{Z}$$

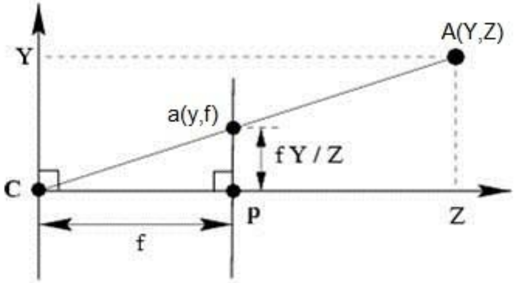
$$\frac{x}{f} = \frac{X}{Z} \longrightarrow x = \frac{fX}{Z}$$

Ref : 1. *"A Flexible New Technique for Camera Calibration", Zhengyou Zhang*
2. *https://in.mathworks.com/help/vision/ug/camera-calibration.html*
3. *https://jordicenzano.name/front-test/2d-3d-paradigm-overview-2011/camera-model/*

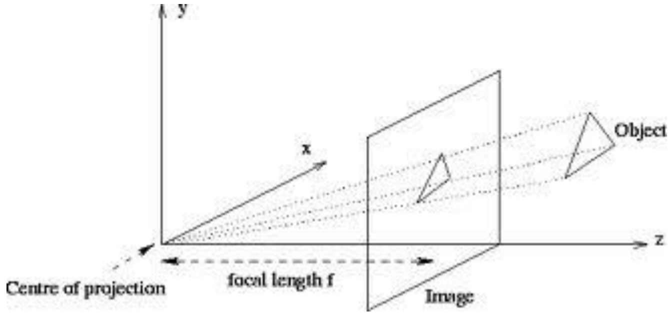
Soliton
Vision for a Better World

Pinhole Camera Model- Another Representation





View - YZ plane



Ref: *Multiple View Geometry in Computer Vision (Second Edition) : Richard Hartley, Andrew Zissermann*

Homogeneous Representation

3D World Point $(X, Y, Z)^T \mapsto (fX/Z, fY/Z)^T$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Homogeneous form of 3D World Point

$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Thin Lens modeling matrix

Soliton
Vision for a Better World

Ref: *Multiple View Geometry in Computer Vision (Second Edition) : Richard Hartley, Andrew Zissermann*

Modeling Camera Sensor Offset

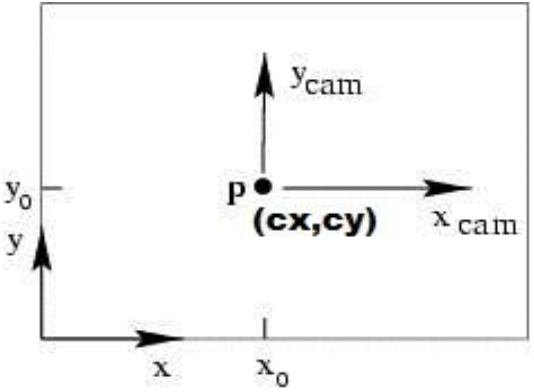


Image Plane

$$(X, Y, Z)^T \mapsto (fX/Z + p_x, fY/Z + p_y)^T$$

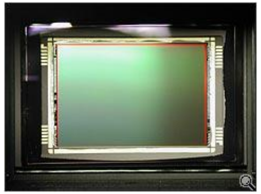
$(p_x, p_y)^T$ principal point

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Zp_x \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Soliton
Vision for a Better World

Ref: *Multiple View Geometry in Computer Vision (Second Edition) : Richard Hartley, Andrew Zissermann*

Modeling Camera Sensor Offset



$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{bmatrix} f_x & & p_x & 0 \\ & f_y & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

- If pixel is skewed

Homogeneous form of point in image plane

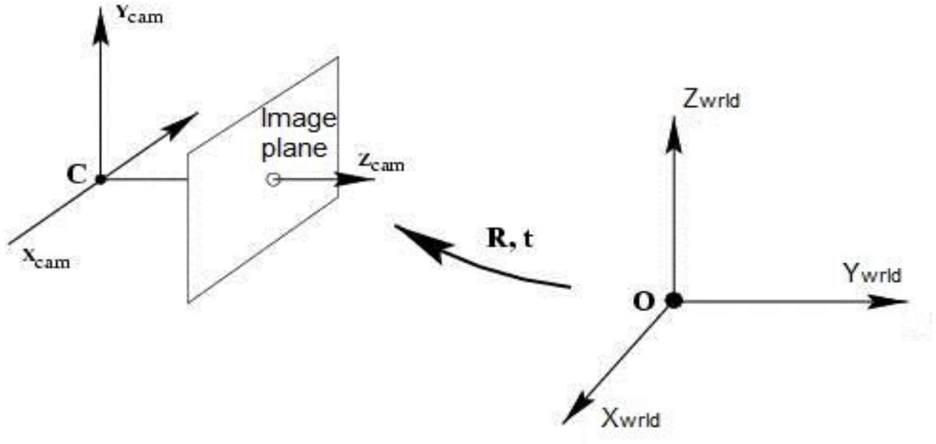
$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{bmatrix} f_x & s & p_x & 0 \\ & f_y & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Homogeneous form of 3D World Point

Ref: *CS766, Fall 2008, Computer Vision, Li Zhang*

Conversion of Coordinate System

- The pinhole model considers object points in camera coordinate system and the real world coordinate system might be different



- Transformation between two co-ordinate system is given by two factors – Rotation and Translation

Soliton
Vision for a Better World

Ref: *Multiple View Geometry in Computer Vision (Second Edition) : Richard Hartley, Andrew Zissermann*

Conversion of Coordinate System

- Point in camera coordinate system to point in world coordinate system

$$P_c = R_{3x3} P_W + T_{3x1}$$

$$\begin{pmatrix} P_c \\ 1 \end{pmatrix} = \begin{bmatrix} R_{3x3} & T_{3x1} \end{bmatrix} \begin{pmatrix} P_W \\ 1 \end{pmatrix}$$

K is 3x3 matrix which defines internal parameters of the camera. It has 5 DOF

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = K_{3x3} \begin{bmatrix} R_{3x3} & T_{3x1} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

[R T] define rotation and translation of camera these are called extrinsic parameters. It has 6 DOF

Soliton
Vision for a Better World

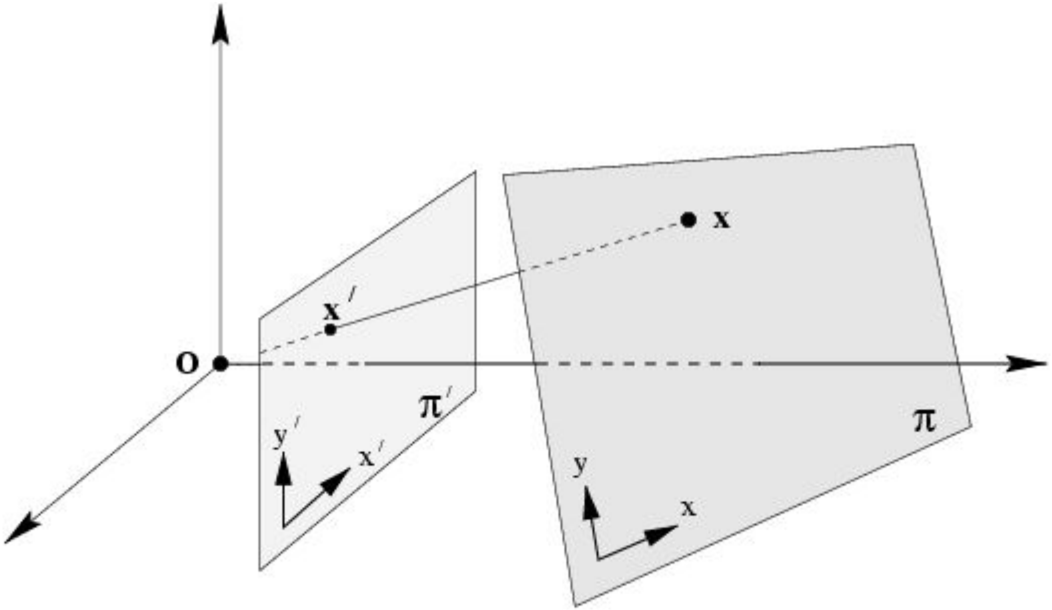
Application of Homography

- This equation can be solved if we know 3D points in real world and its corresponding 2D points in image
- Error chances are high when we use 3D points and ‘ease of use’ is low
- If all points are in single plane, it will become plane to plane transformation eliminating one of the dimension

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = K_{3x3} \begin{bmatrix} R_{3x3} & T_{3x1} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix}$$

Soliton
Vision for a Better World

Mapping between planes



Projection from one plane to another may be expressed by

x'=Hx

Ref: *Multiple View Geometry in Computer Vision (Second Edition) : Richard Hartley, Andrew Zissermann*

Application of Homography

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = K_{3x3} \begin{bmatrix} R_{3x2} & T_{3x1} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

Plane to plan transformation (H)

$$p_{cam} = H_{3x3} P_{World}$$

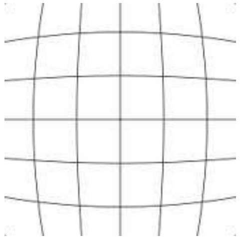
$$H = K_{3x3} \begin{bmatrix} R_{3x2} & T_{3x1} \end{bmatrix}$$

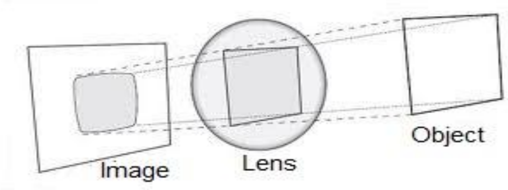
Given set of corresponding points in real world plane (checkerboard) and point in image we can find the H and decompose H into K, R and T

Lens effect

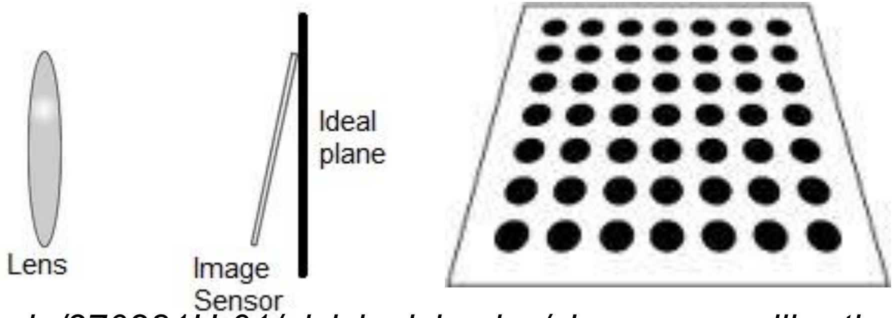
Camera model doesn't consider lens effects

- Lens – to focus light and converge
- Distortions
    - Radial Distortion – shape of lens





    - Tangential Distortion – image sensor not parallel to lens



Soliton
Vision for a Better World

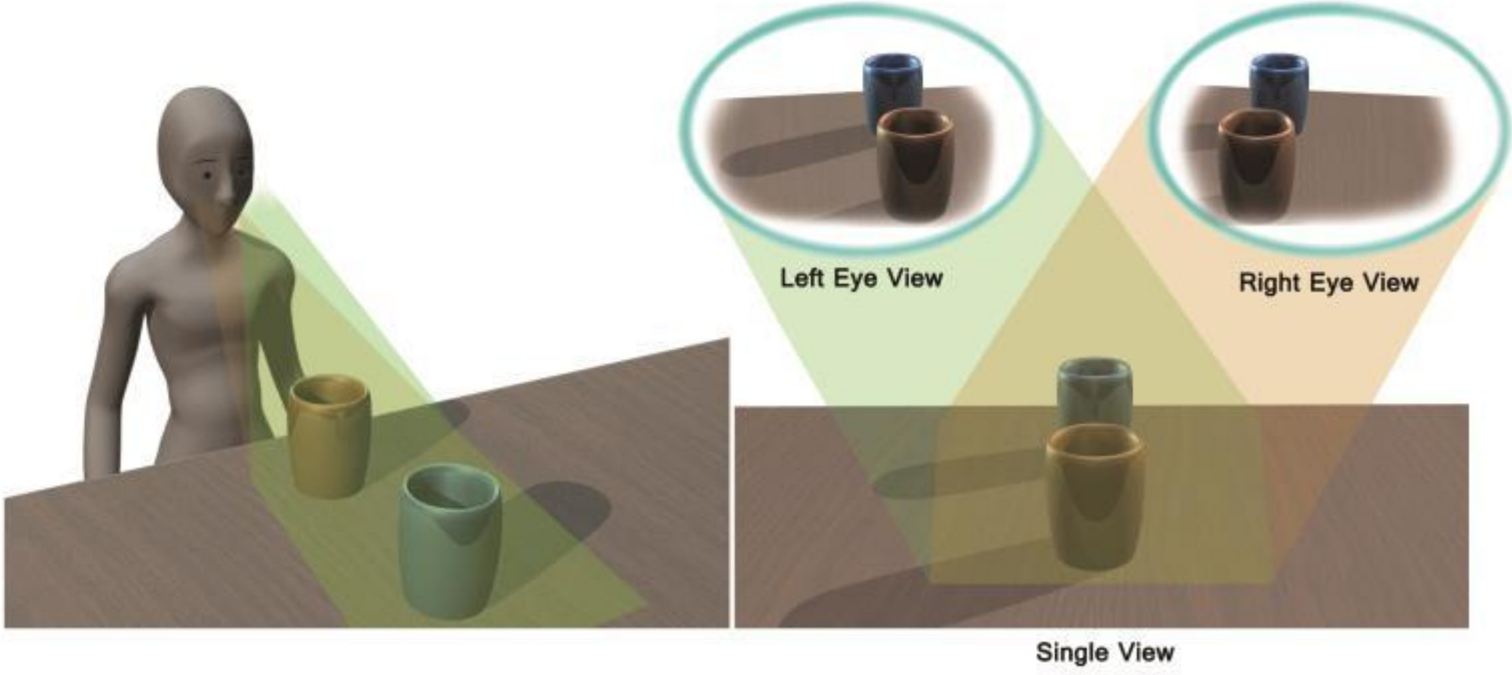
***Ref:**http://zone.ni.com/reference/en-XX/help/370281U-01/nivisionlvbasics/choose_a_calibration_type/*

Overview of Camera Calibration

- Object points - known object plane
- Image points - Detection of feature points in image
- Homography matrix using correspondence between image points and object points.
- Decompose homography matrix to K, R and T
- Follow the above procedure for large samples
- Result : Intrinsic matrix K

Soliton
Vision for a Better World

Human Eye - Stereo vision

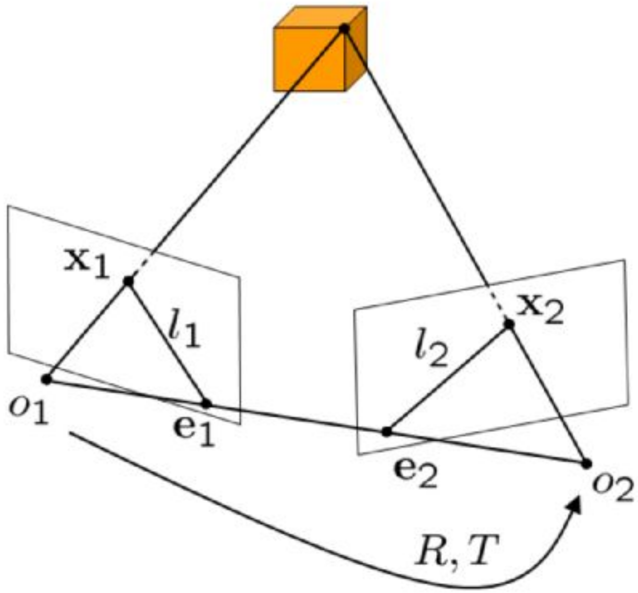


Stereo Camera

Goal: Estimate camera motion and 3D scene structure from two views.

Epipolar Geometry

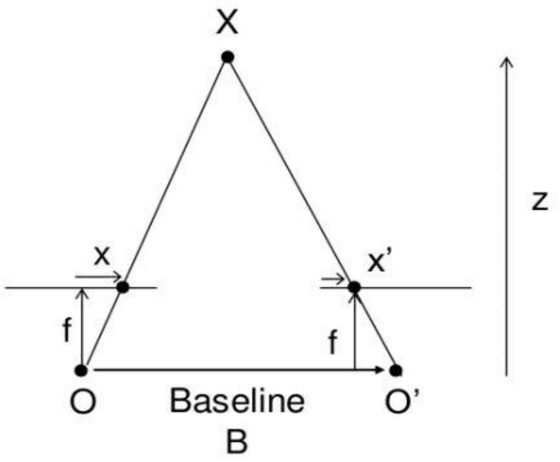
- The projections of a point **X** onto the two images are denoted by $\mathbf{x}_1$ and $\mathbf{x}_2$
- The **optical centers** of each camera are Denoted by $\mathbf{o}_1$ and $\mathbf{o}_2$
- The intersections of the line ($\mathbf{o}_1$, $\mathbf{o}_2$) with each image plane are called the **epipoles** $\mathbf{e}_1$ and $\mathbf{e}_2$
- The intersections between the **epipolar plane (**$\mathbf{o}_1$ **,** $\mathbf{o}_2$ **, X)** and the image planes are called **epipolar lines** $\mathbf{l}_1$ **and** $\mathbf{l}_2$
- There is one epipolar plane for each 3D point **X**



Depth Map Calculation from Stereo Images

- Stereo setup diagram contains equivalent triangles. Writing their equivalent equations will yield us following result:

$$disparity = x - x' = \frac{Bf}{Z}$$



- x and x' are the distance between points in image plane corresponding to the scene point 3D and their camera center. B is the distance between two cameras (which we know) and f is the focal length of camera (already known). So in short, above equation says that the depth of a point in a scene is inversely proportional to the difference in distance of corresponding image points and their camera centers.

Soliton
Vision for a Better World

The Epipolar Constraint

- We know that $\mathbf{x}_1$ (in homogeneous coordinates) is the projection of a 3D point $\mathbf{X}$. Given known camera parameters ($\mathbf{K} = \mathbf{1}$) and no rotation or translation of the first camera, we merely have a projection with unknown depth $\boldsymbol{\lambda}_1$. From the first to the second frame we additionally have a camera rotation $\mathbf{R}$ and translation $\mathbf{T}$ followed by a projection. This gives the equations:

$$\lambda_1 \mathbf{x}_1 = \mathbf{X}, \qquad \lambda_2 \mathbf{x}_2 = \mathbf{R}\mathbf{X} + \mathbf{T}.$$

- Inserting the first equation into the second and simplifying it, we get following equation: $\boxed{(\mathbf{x}_2)^T[\mathbf{T}]_x \mathbf{R}\,\mathbf{x}_1 = 0}$ $[\mathbf{T}]_x$ = translation skew-symmetric 3x3 matrix
- This provides a relation between the 2D point coordinates of a 3D point in each of the two images and the camera transformation parameters.

Soliton
Vision for a Better World

Essential Matrix and Fundamental Matrix

- In the previous equation the original 3D point coordinates have been removed. The matrix $\mathbf{E = [T]_x R \in R^{3\times3}}$ is called the **essential matrix**. The **epipolar constraint** is also known as **essential constraint** or **bilinear constraint**.
- Geometrically, this constraint states that the three vectors $\mathbf{o_1X}$, $\mathbf{o_2\,o_1}$ and $\mathbf{o_2X}$ form a plane, i.e. the triple product of these vectors (measuring the volume of the parallelepiped) is zero:

$$\mathbf{volume = (x_2)^T\,(T\ X\ Rx_1\,) = (x_2)^T[T]_xR\ x_1 = 0}$$

- By transforming all image coordinates **x\`** with the inverse calibration matrix $\mathbf{K^{-1}}$ into metric coordinates **x**, we obtain the epipolar constraint for uncalibrated cameras: $\mathbf{(x`_2)^T K^{-T}[T]_x RK^{-1}x`_1 = 0} \quad \Leftrightarrow \quad \mathbf{x`_2\,F\,x`_1 = 0}$

Soliton
Vision for a Better World

The Eight-Point Linear Algorithm

- First we rewrite the epipolar constraint as a scalar product in the elements of the matrix **E** and the coordinates of the points $\mathbf{x}_1$ and $\mathbf{x}_2$. Let

  $$\mathbf{E^s} = (e_{11}, e_{21}, e_{31}, e_{12}, e_{22}, e_{32}, e_{13}, e_{23}, e_{33})^T \in \mathbf{R}^9$$

  be the vector of elements of **E** and $\mathbf{x_i} = (x_i, y_i, z_i)$

  $$\mathbf{a} = (x_1x_2, x_1y_2, x_1z_2, y_1x_2, y_1y_2, y_1z_2, z_1x_2, z_1y_2, z_1z_2) \in \mathbf{R}^9$$

- Then the epipolar constraint can be written as:

  $$(\mathbf{x}_2)^T\mathbf{E}\mathbf{x}_1 = \mathbf{a}^T\mathbf{E}^s = 0$$

- For n point pairs, we can combine this into the linear system and solve for $\mathbf{E^s}$
- Recover the displacement from the essential matrix decomposition into four possible solutions for rotation and translation.

Soliton
Vision for a Better World

Essential Matrix Decomposition

- The space of all essential matrices is called the **essential space**:

  $$\mathbf{e} \equiv \{[T]_x R \mid R \in \text{Special Orthogonal Matrix 3x3},\ T \in \mathbb{R}^3\} \subset \mathbb{R}^{3\times3}$$

- A nonzero matrix $\mathbf{E} \in \mathbb{R}^{3\times3}$ is an essential matrix if and only if $\mathbf{E}$ has a singular value decomposition (SVD) $\mathbf{E} = \mathbf{U\Sigma V^T}$ with

  $\Sigma = \text{diag}\{\sigma, \sigma, 0\}$ for some $\sigma > 0$ and $\mathbf{U, V} \in SO(3)$.

- Theorem (Pose recovery from the essential matrix): There exist exactly two relative poses $(\mathbf{R}, \mathbf{T})$ with $\mathbf{R} \in SO(3)$ and $\mathbf{T} \in \mathbb{R}^3$ corresponding to an essential matrix $\mathbf{E} \in \mathbf{e}$. For $\mathbf{E} = \mathbf{U\Sigma V^T}$ we have:

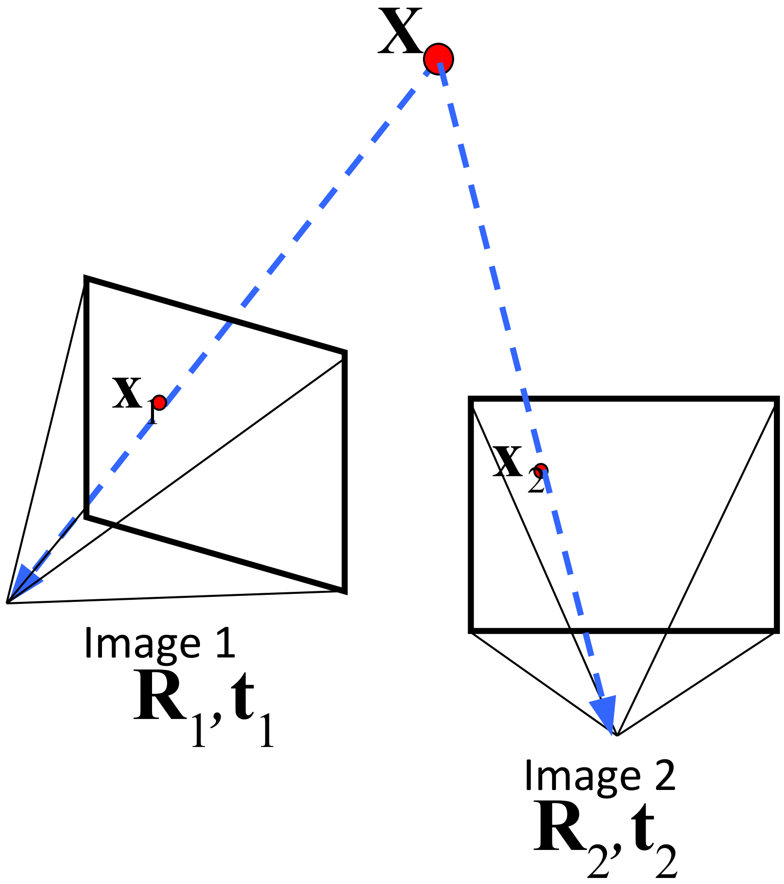
  $$([T]_{x1}, R_1) = UR_Z(+\pi/2)\Sigma U^T,\ U(R_Z)^T(+\pi/2)V^T, \qquad (1)$$
  $$([T]_{x2}, R_2) = UR_Z(-\pi/2)\Sigma U^T,\ U(R_Z)^T(-\pi/2)V^T, \qquad (2)$$

- In general, only one of these gives meaningful (positive) depth values.

Soliton
Vision for a Better World

Triangulation from two views

- Estimate **R** and **T** from 4 possible solutions (select **R** and **T** that when substituted provides the positive depth)
- Use R and T to recover the depth of the 3D points and this give use the all 3D point corresponding to the each corresponding matches in the two images



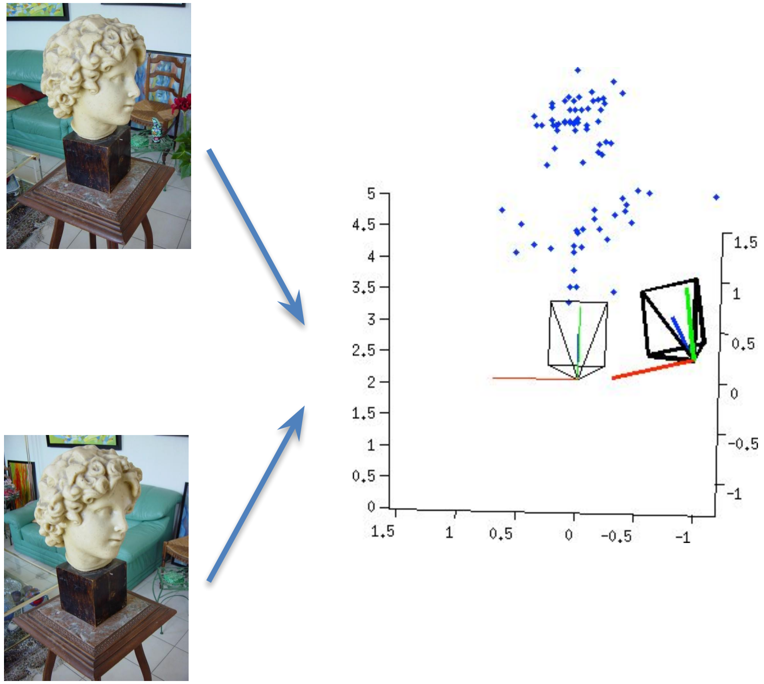
Soliton
Vision for a Better World

A Basic Reconstruction Algorithm

- We have seen that the 2D-coordinates of each 3D point are coupled to the camera parameters R and T through an epipolar constraint. In the following, we will derive a 3D reconstruction algorithm which proceeds as follows:
- We assume that we are given a set of corresponding points in two frames taken with the same camera from different vantage points.
- We assume that the scene is static, i.e. none of the observed 3D points moved during the camera motion
- **Recover the essential matrix E** from the epipolar constraints associated with a set of point pairs.
- **Extract the relative translation and rotation** from the essential matrix **E.**
- **Triangulate from using R and T** to get 3D points

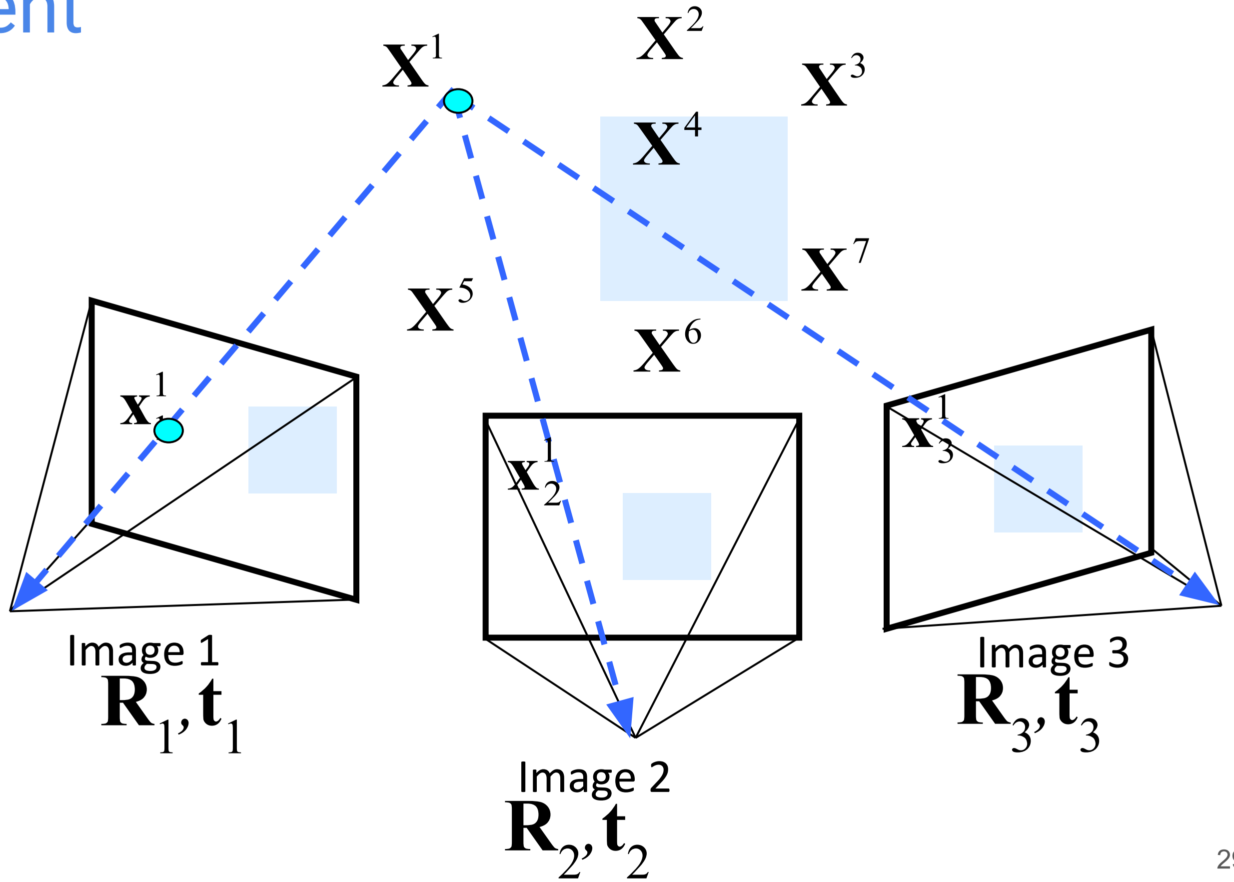
Reconstruction from two views

- Reconstructed point cloud from two views



Bundle Adjustment

- Multiple 3D points as seen from multiple viewpoints
- Same points is visible in all three views



Soliton
Vision for a Better World

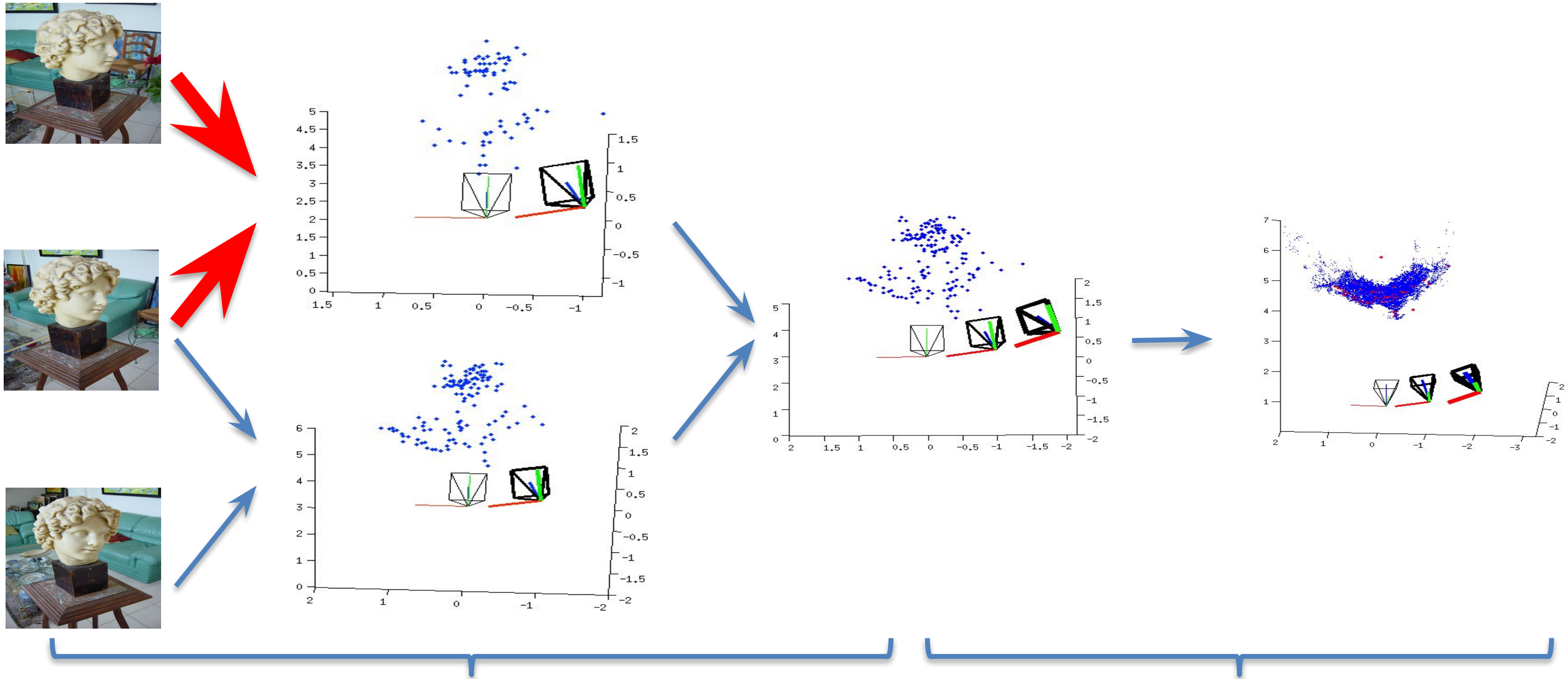
Bundle Adjustment

	Point 1	Point 2	Point 3
Image 1	$\mathbf{x}_1^1 = \mathbf{K}\left[\mathbf{R}_1 \mid \mathbf{t}_1\right]\mathbf{X}^1$	$\mathbf{x}_1^2 = \mathbf{K}\left[\mathbf{R}_1 \mid \mathbf{t}_1\right]\mathbf{X}^2$	
Image 2	$\mathbf{x}_2^1 = \mathbf{K}\left[\mathbf{R}_2 \mid \mathbf{t}_2\right]\mathbf{X}^1$	$\mathbf{x}_2^2 = \mathbf{K}\left[\mathbf{R}_2 \mid \mathbf{t}_2\right]\mathbf{X}^2$	$\mathbf{x}_2^3 = \mathbf{K}\left[\mathbf{R}_2 \mid \mathbf{t}_2\right]\mathbf{X}^3$
Image 3	$\mathbf{x}_3^1 = \mathbf{K}\left[\mathbf{R}_3 \mid \mathbf{t}_3\right]\mathbf{X}^1$		$\mathbf{x}_3^3 = \mathbf{K}\left[\mathbf{R}_3 \mid \mathbf{t}_3\right]\mathbf{X}^3$

- A valid solution for $\mathbf{R_1|t_1}$, $\mathbf{R_2|t_2}$ and $\mathbf{R_3|t_3}$ will be the one the minimize the reprojection error of the 3d points from multiple views:

$$\min \Sigma_i \Sigma_j ((x_i)^j - K[R_i|T_i]X^j)^2$$ Optimization problem

Bundle Adjustment

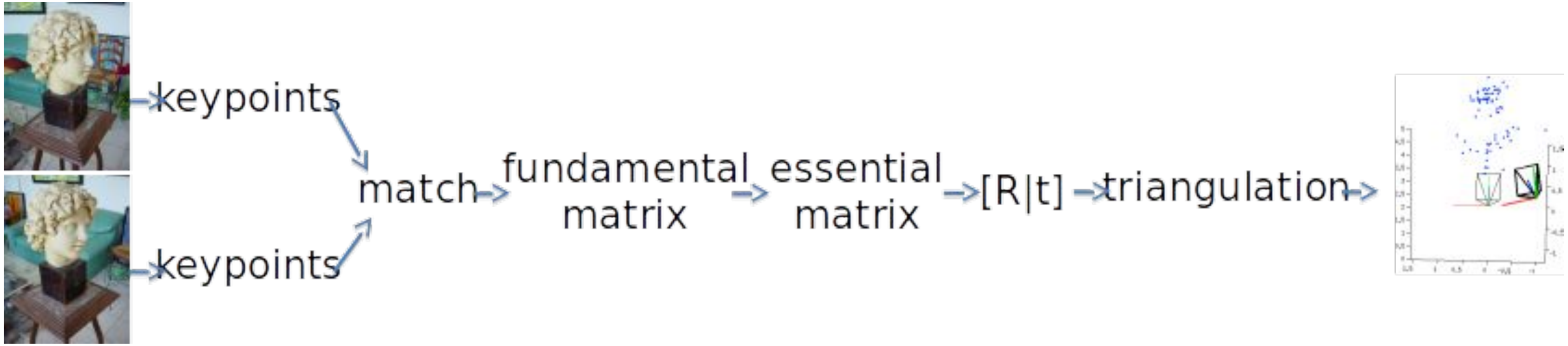


Structure from Motion (SFM)

Multi-view Stereo (MVS)

3D Reconstruction Steps

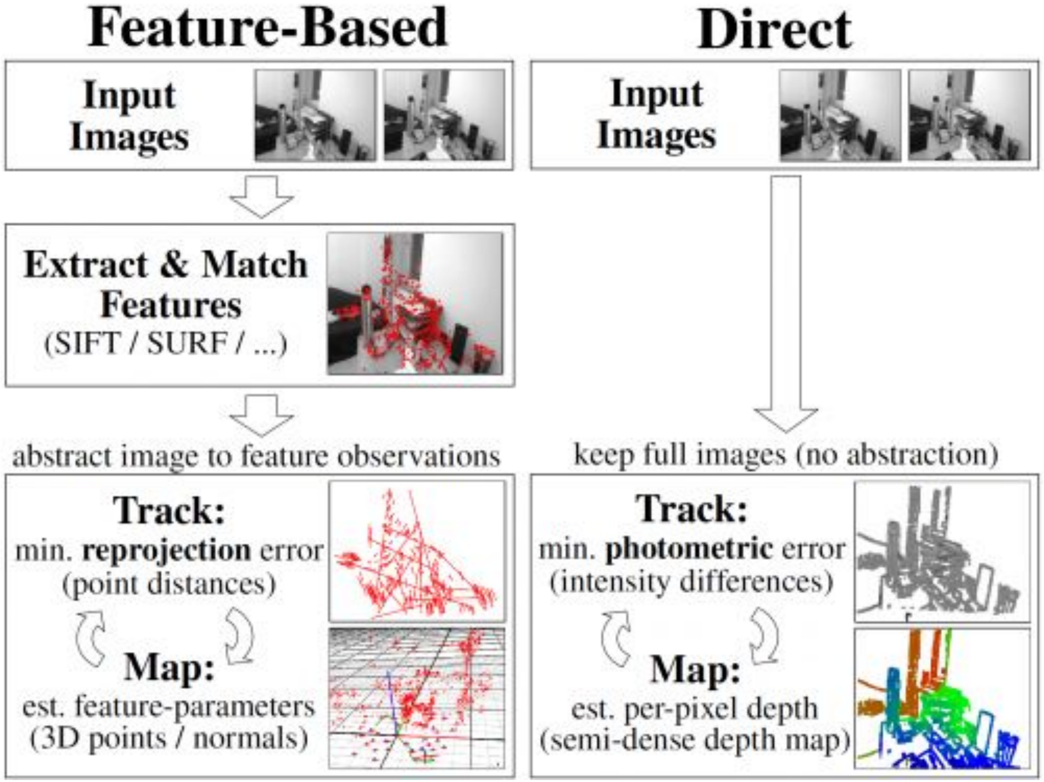
-



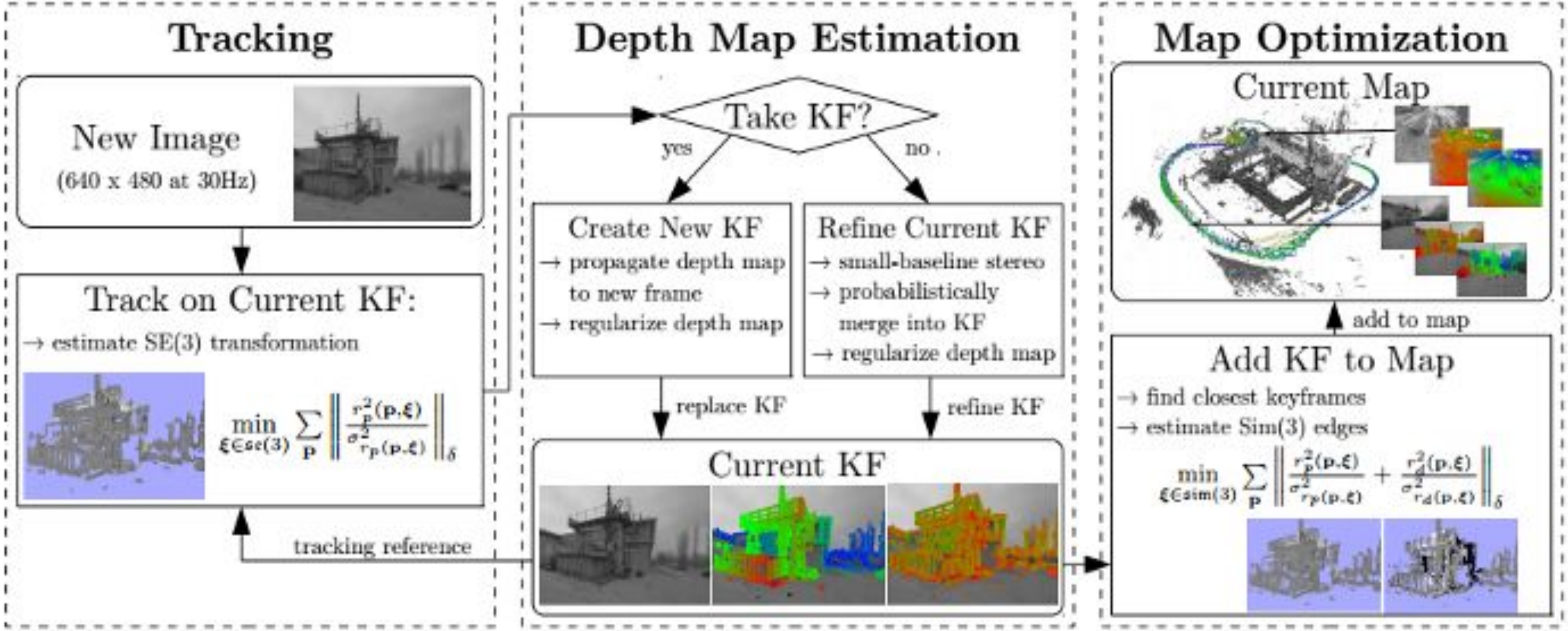
SLAM introduction

- Localization - Determine the pose given a map
- Mapping - Generate a map when pose is known
- SLAM - key steps
  - Defined by an arbitrary coordinate system (initial pose)
  - Generate a map using sensors, and at the the same time compute pose
  - Map errors and pose estimate are correlated

SLAM algorithm



LSD SLAM



Tracking
New Image
(640 x 480 at 30Hz)
Track on Current KF:
→ estimate SE(3) transformation
$\min_{\xi \in se(3)} \sum_{\mathbf{p}} \left\| \frac{r_p^2(\mathbf{p},\xi)}{\sigma^2_{r_p(\mathbf{p},\xi)}} \right\|_\delta$
tracking reference
Depth Map Estimation
Take KF?
yes
no
Create New KF
→ propagate depth map to new frame
→ regularize depth map
Refine Current KF
→ small-baseline stereo
→ probabilistically merge into KF
→ regularize depth map
replace KF
refine KF
Current KF
Map Optimization
Current Map
add to map
Add KF to Map
→ find closest keyframes
→ estimate Sim(3) edges
$\min_{\xi \in sim(3)} \sum_{\mathbf{p}} \left\| \frac{r_p^2(\mathbf{p},\xi)}{\sigma^2_{r_p(\mathbf{p},\xi)}} + \frac{r_d^2(\mathbf{p},\xi)}{\sigma^2_{r_d(\mathbf{p},\xi)}} \right\|_\delta$

Advice on Applying ML / DL

- Case studies from Anthill Inside ppt

Soliton
Vision for a Better World

Disciplined Machine Learning

-

Soliton
Vision for a Better World

GAN

- StackGAN
- Conditional GAN
- InfoGAN
- Self Attention GAN
- Image-to-Image Translation with Conditional Adversarial Networks

Soliton
Vision for a Better World

Attention Networks

- Compositional Attention Networks
- Hierarchical Recurrent Attention Network for Response Generation

Soliton
Vision for a Better World

Session IV
Practical DL

Agenda

- ➔ Real World Problem Definition
- ➔ AVA Dataset
- ➔ Steps involved

http://lodgiq.com/improved-pricing-with-machine-learning/

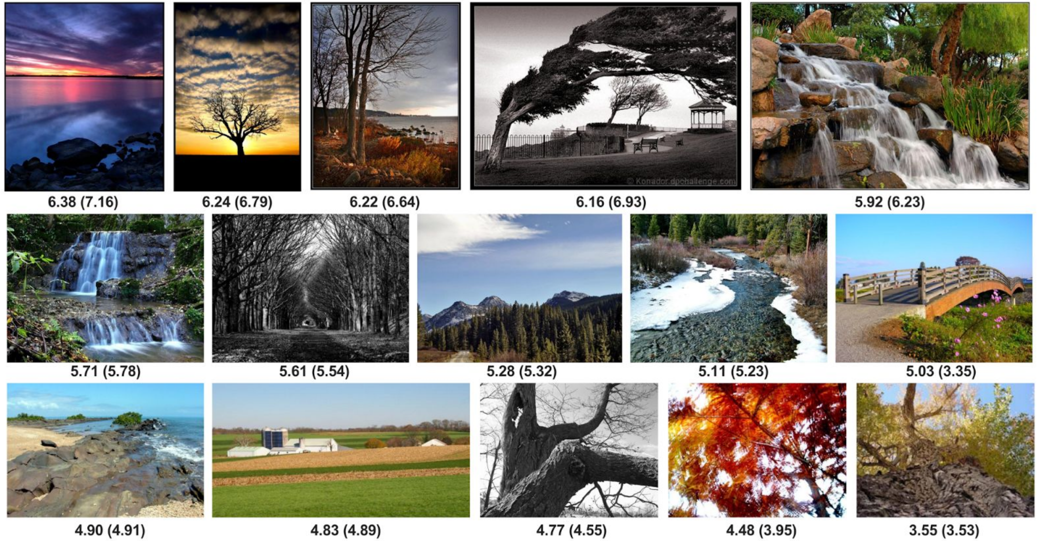
Soliton
Vision for a Better World

Aesthetic Scoring

Problem statement: Given an image, Rate it based on the Aesthetics of the Image

AVA Dataset

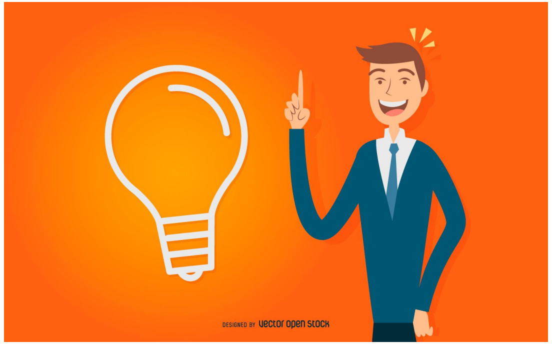
- Large scale aesthetics dataset
- Each image is scored between 0 to 10 by multiple human reviewers



Img ref-https://ai.googleblog.com/2017/12/introducing-nima-neural-image-assessment.html

AVA Dataset

The most straightforward idea



- Try 10 class classification
- How to get a label for each image?
  - Max: choose the most voted score
  - Average: Calculate the average of all assigned scores
- How to sample data?
  - Sample 10k images for each class for training and keep the rest for testing.

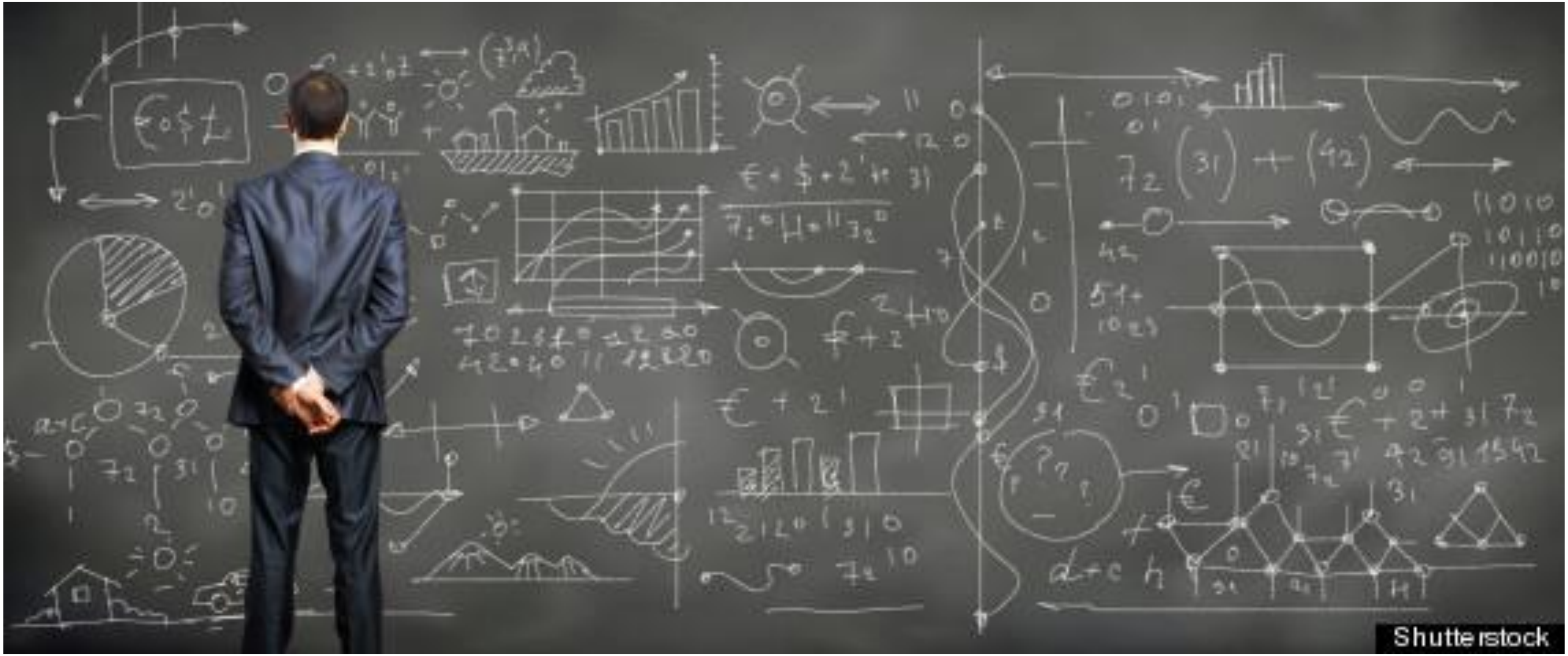
Img Ref-https://www.vexels.com/vectors/preview/78830/idea-man-drawing

What architecture to choose?

- Can we try our own network?
- Can we try out a ready made architecture like ResNet, AlexNet or GoogleNet?
- Larger data → deeper architecture
- Smaller data → simple and shallow architecture

Image ref- https://isha.sadhguru.org/in/en/wisdom/article/confusion-and-clarity-on-the-spiritual-path

Why is the accuracy low?



Img ref-http://leesclassroom.global2.vic.edu.au/2014/03/19/maths-problem-solving-2/

Class imbalance

- **Class imbalance:** Unequal data for all classes. The model is biased against or towards certain classes.
- Training is biased and hence accuracy is also biased
- **Good practices:** Always visualise the data before spending too much time in training.

Poor progress so far. What else is the problem? Let’s keep moving.

Img Ref -http://www.panditrajeevraosharma.com/business-problem.html

Critical analysis of the loss function

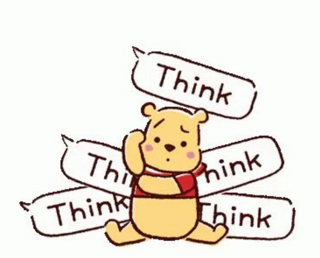
- Let's take a simple example to decode the problem with the loss function.
  - Example 1: Let's say the true label was 1 - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  - But the predicted label had this probability distribution - [0.05, 0, 0, 0, 0, 0, 0.95, 0, 0, 0]
  - Binary cross entropy loss = $-\sum y_i \log y'_i$

    = -1 * log 0.05 = 1.31

  - Example 2: Let's say the true label was 1 - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  - But the predicted label had this probability distribution- [.05, 0.95, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  - Binary cross entropy loss = $-\sum y_i \log y'_i$

    = -1 * log 0.05 = 1.31

- Can you figure out the problem?



Img Ref - https://tenor.com/search/think-think-think-winnie-the-pooh-gifs

Is there a better loss function?

- **Weighted L2 loss function**
  - Example : Let's say the true label was 1 - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  - But the predicted label had this probability distribution - [0.05, 0, 0, 0, 0, 0, 0.95, 0, 0, 0]
  - Weighted L2 loss = $\sum w_i * |y_i - y'_i|$ where w_i = | G.T index- i | + 1

    = 1 * 0.95 + 6 * 0.95 = 6.65

  - Example: Let's say the true label was 1 - [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  - But the predicted label was 2 - [.05, 0.95, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  - Weighted L2 loss = $\sum w_i * |y_i - y'_i|$ where w_i = | G.T index - i | + 1

    = 1 * 0.05 + 2 * 0.95 = 1.95

- Let's improve our model with this refined loss function

Img Ref - http://www.monday-8am.com/getting-better-with-age/

Let’s visualise the data.

- Visualising data gives us some intuitions and exposes shortcomings of the current model
- Is my train data representative of the real time data.

Img Ref - http://www.stbrigidsms.wa.edu.au/newsletter/view/1/101-week-7-term-1

Let’s visualise the data.

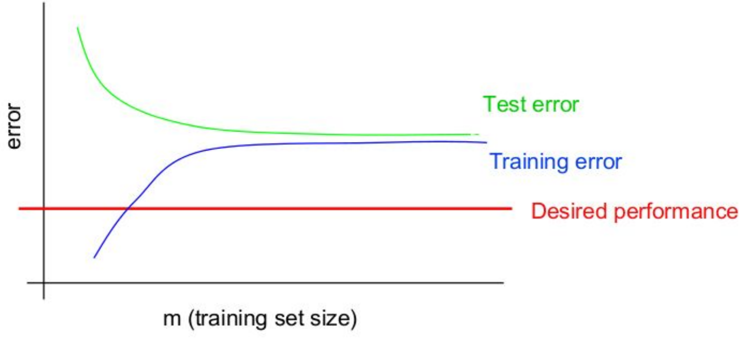
- Visualising data gives us some intuitions and exposes shortcomings of the current model
- Is my train data representative of the real time data.

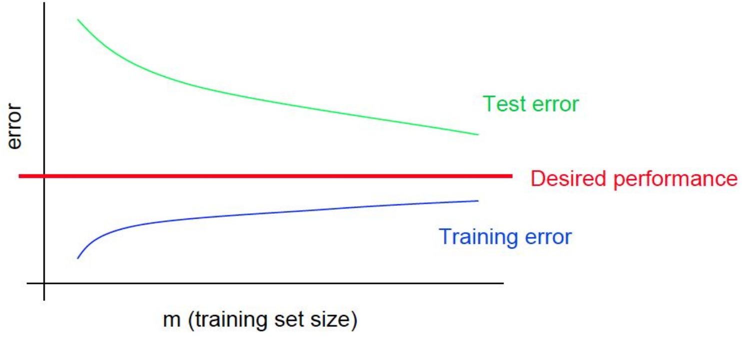
Some observations and questions:-

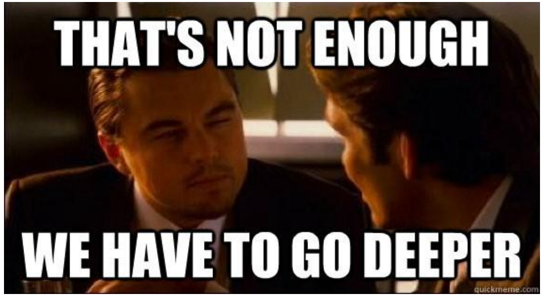
- Aspect ratio of the image plays an important role
- Can we group classes together and reduce the number of classes?

Img Ref - http://www.stbrigidsms.wa.edu.au/newsletter/view/1/101-week-7-term-1

Seeing familiar curves. Lets try a deeper model



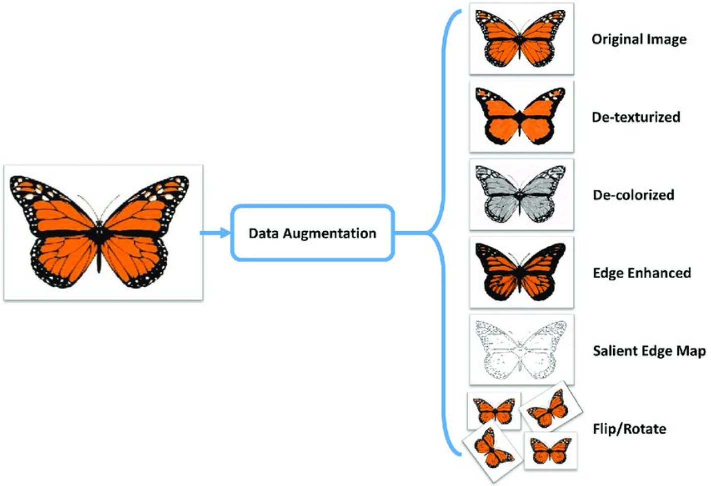




Credits: cs229, Andrew Y. Ng
Img Ref - https://medium.com/@bakiiii/microsoft-presents-deep-residual-networks-d0ebd3fe5887

But is this data enough? If not, Augment the data

- What transformations on my image leave the label of the image unchanged?
- Apply all those transformations to augment the limited data set.



Img Ref - https://www.researchgate.net/figure/Data-augmentation-using-semantic-preserving-transformation-for-SBIR_fig2_319413978

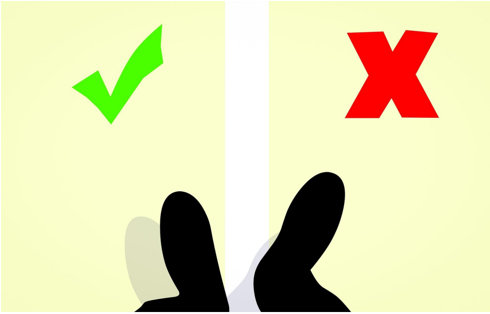
Can we add metadata?

- Tagging images with useful metadata could improve the accuracy
- What tags could be useful for AVA dataset??
  - Nature of the scene: Marriage, Playground, Forest etc..
  - Number of people in the image

Img Ref- https://www.dpconline.org/handbook/organisational-activities/metadata-and-documentation

Changing metrics

- Is our final accuracy metric in line with our training objective?
- Calculating accuracy in terms of predicting a correct label is misleading.
- Calculating how far is the predicted label from the true label may give a better measure.
- Better ways of generating data: Which of these statements is better for data collection?
    - Rate this image in a scale of 1-10.
    - How is the image? Excellent, good, average, below average, poor.



https://www.anirudhsethireport.com/visualise-and-be-motivated/

Overgeneralization

- Am I trying to solve a more complex problem than what is actually required?
- Does solving the more complex problem add more business value?
- An over generalising problem statement:
    - Which picture has a great aesthetics with story telling value?

https://sites.google.com/a/aguafria.org/block-2-logical-fallacies/home/overgeneralization-fallacy

Advice on Applying Machine Learning: War Stories

Sumod K Mohan

AutoInfer

Soliton

Anti-pattern 0 : Lott’ Data & AI Magic Sparkle

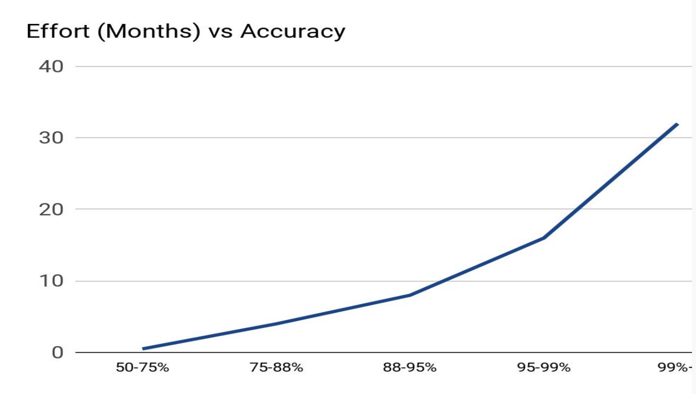
- Let’s use AI <insert favorite jargon instead>, everyone’s using it
  - Trivial: Lets use AI suggest to do spell check, fit a line etc
  - Complex: Chatbot to converse on any given topic
    Google’s Duplex does it: Narrow ability in specific skills
  - Complicated: Replace Doctor’s (text, speech, viewing images, emotions etc)

- Lott’ Data: Magic Sparkle of AI: Sprinkle and forget
- Can’t we use txfer learning/unsupervised/RL: Nuances matter
- Low Data: Augment Data, Can’t you use GAN’s ?
- Recent paper that solves prob y, why can’t we use it for x

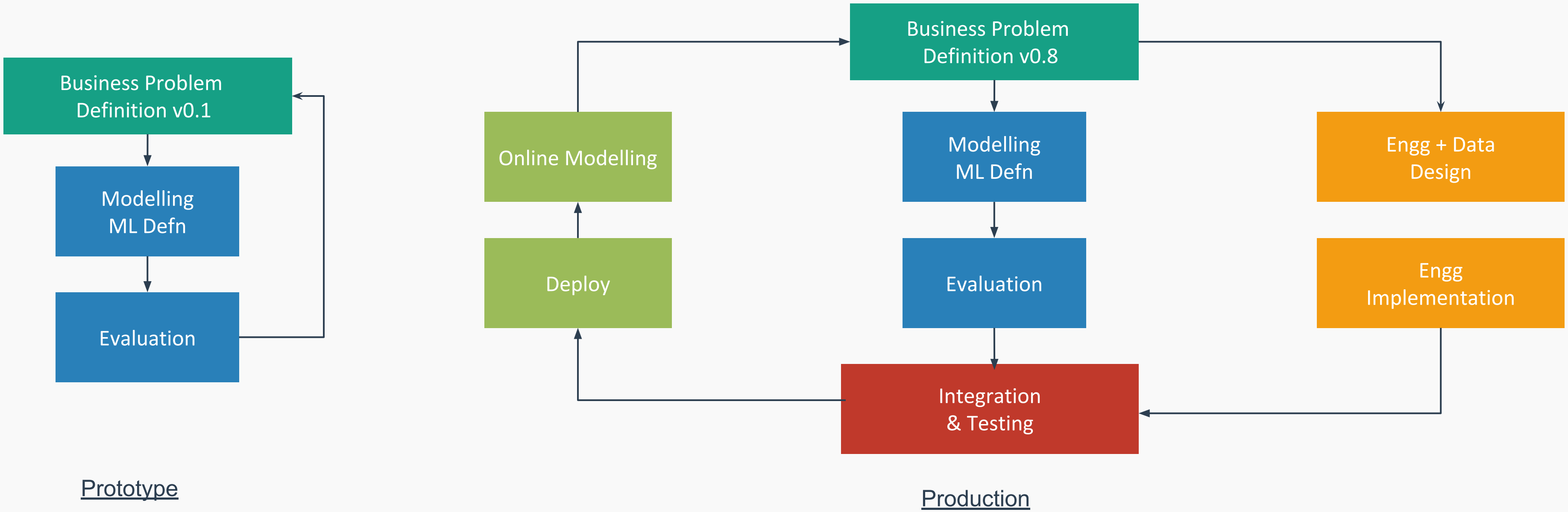
Anti-pattern 0 : Lott' Data & AI Magic Sparkle

- Gets your hands dirty & See beyond jargons
- Hold ML Sessions/Attend meetups to get a hang of nuances
- Trust the people whose hands are dirty but verify solving right problem

  " ..in research, in general the people that are doing it are in the best position to evaluate it, not the people that are supervising it ...": Robert Noyce, Co-Founder, Intel
- Hopefully this talk will help to make better decisions
- Understand ML/DL Software Lifecycle (next)



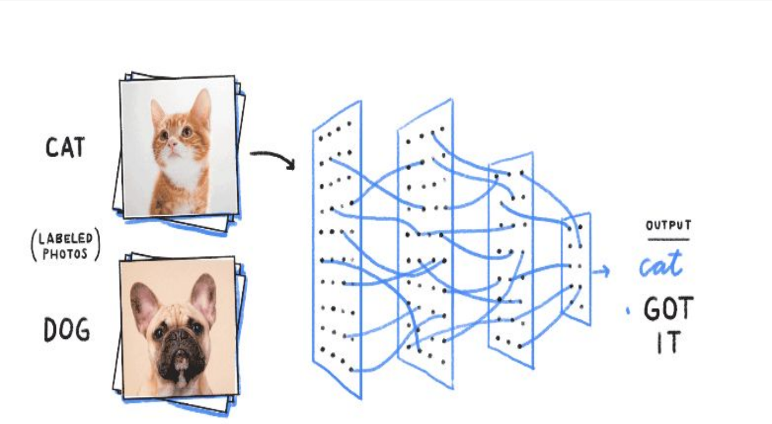
Anti-pattern 0 : Lott' Data & AI Magic Sparkle



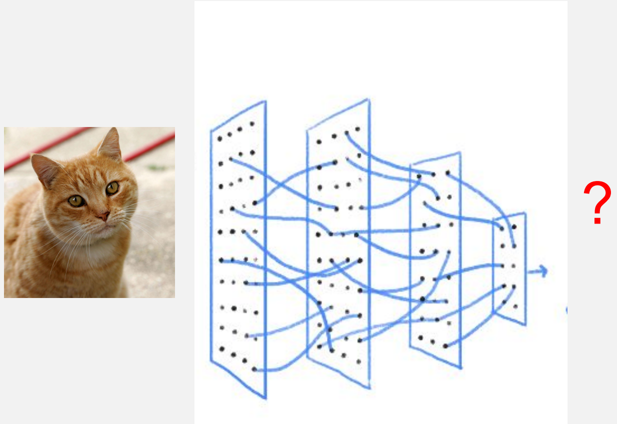
Anti-pattern 1 : Garbage in - Garbage out

Typical DL/ML System (supervised)

Train



Test



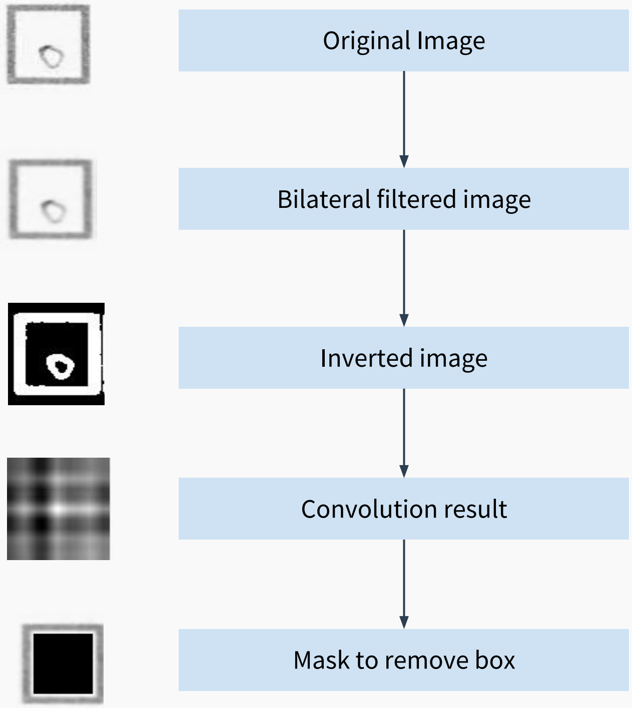
Anti-pattern 1 : Garbage in - Garbage out

Symptoms

- Added new data, system gone haywire
- Model on knife-edge, minor tweak all hell breaks loose
- Long Repeated Iteration Loops for Developing models
  - Initially took 4 months, more data again took 5 months, more data...
- Model gives crazy results every now and then

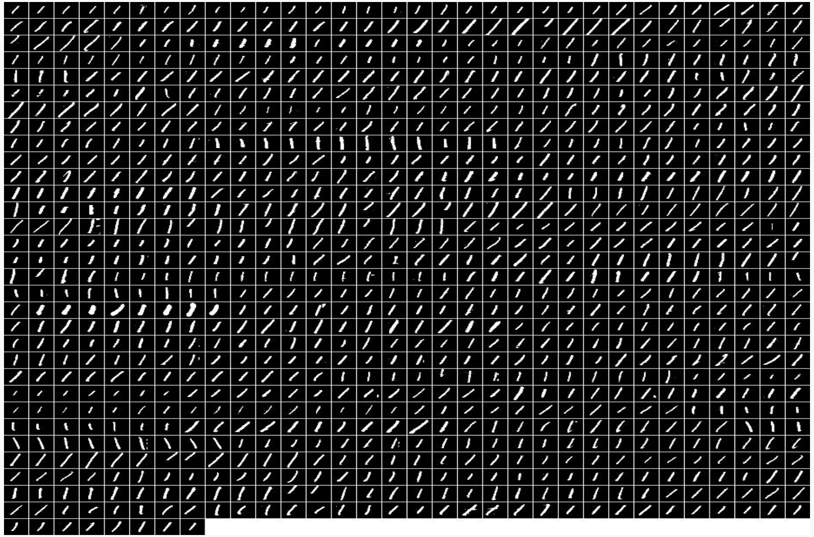
Anti-pattern 1 : Garbage in - Garbage out

- Look at your data
  - Kosher: Nothing off in Input and Output
  - Do a simple walk through each stage
- Can you overfit your model (98-99%+ training)
- Visualize label/output distribution: Is it nearly equal, if not then handling uneven classes
- Improving Beyond (large data issues)
  - Tools to effectively view large quantity of data

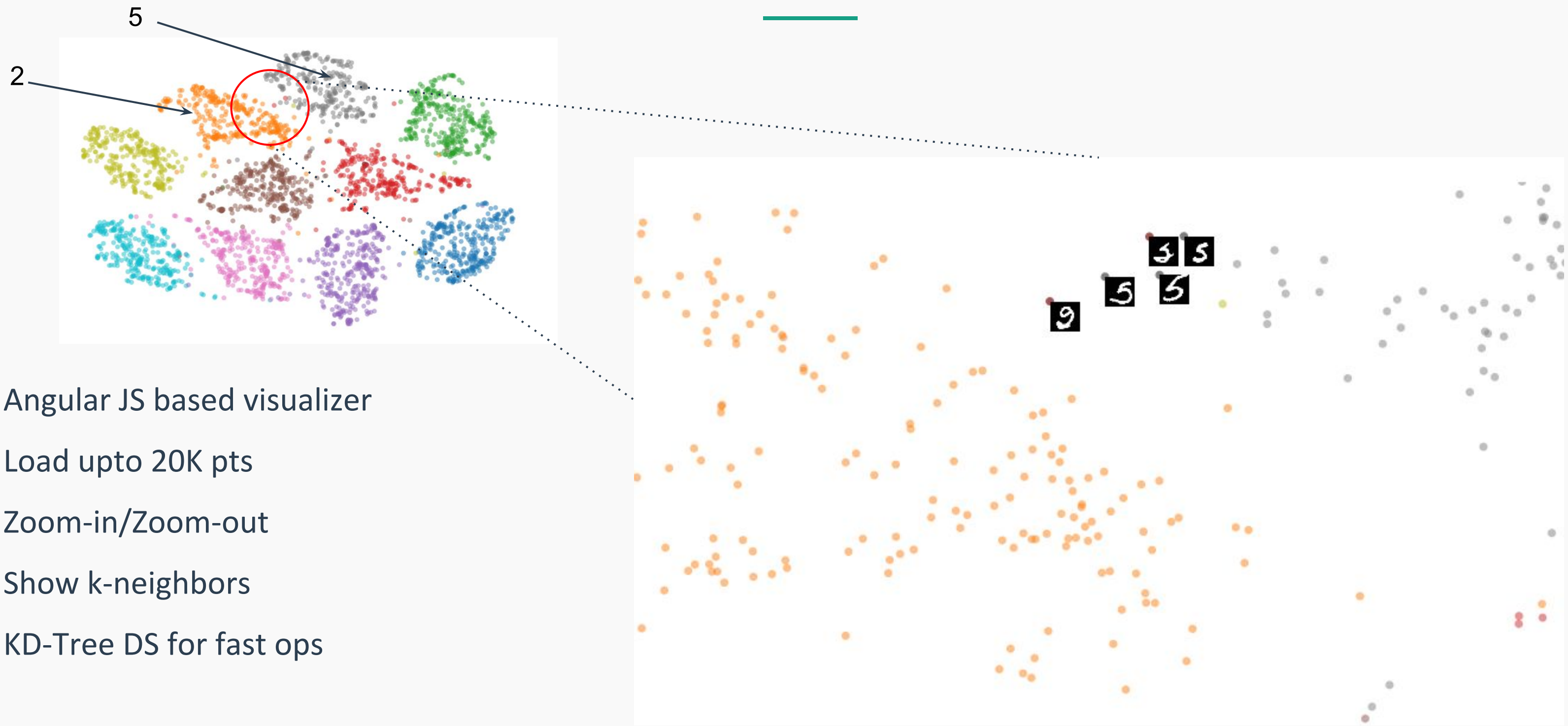


Anti-pattern 1 : Garbage in - Garbage out

- Look at your data
  - Kosher: Nothing off in Input and Output
  - Do a simple walk through each stage
- Can you overfit your model (98-99%+ training)
- Visualize label/output distribution: Is it nearly equal, if not then handling uneven classes
- Improving Beyond (large data issues)
  - Tools to effectively view large quantity of data



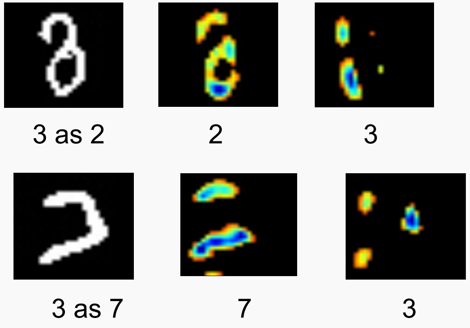
Data Visualization



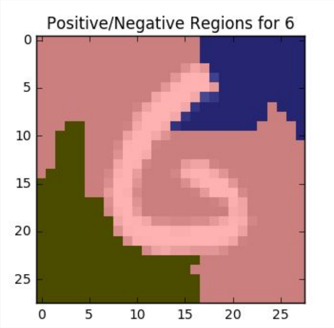
- Angular JS based visualizer
- Load upto 20K pts
- Zoom-in/Zoom-out
- Show k-neighbors
- KD-Tree DS for fast ops

Anti-pattern 1 : Garbage in - Garbage out

- Look at your data
  - Kosher: Nothing off in Input and Output
  - Do a simple walk through each stage
- Can you overfit your model (98-99%+ training)
- Visualize label/output distribution: Is it nearly equal, if not then handling uneven classes
- Improving Beyond (large data issues)
  - Tools to effectively view large quantity of data
  - Look at critical regions & Inspect your model: CAM/LIME
  - Is it really confusing or learning unrelated patterns (Bach vs Mozart)
- Synthesize or Augment or Simulate
  - GANs
  - Simulate



Class Activation Maps



LIME: Locally Interpretable Model Agnostic Explanations

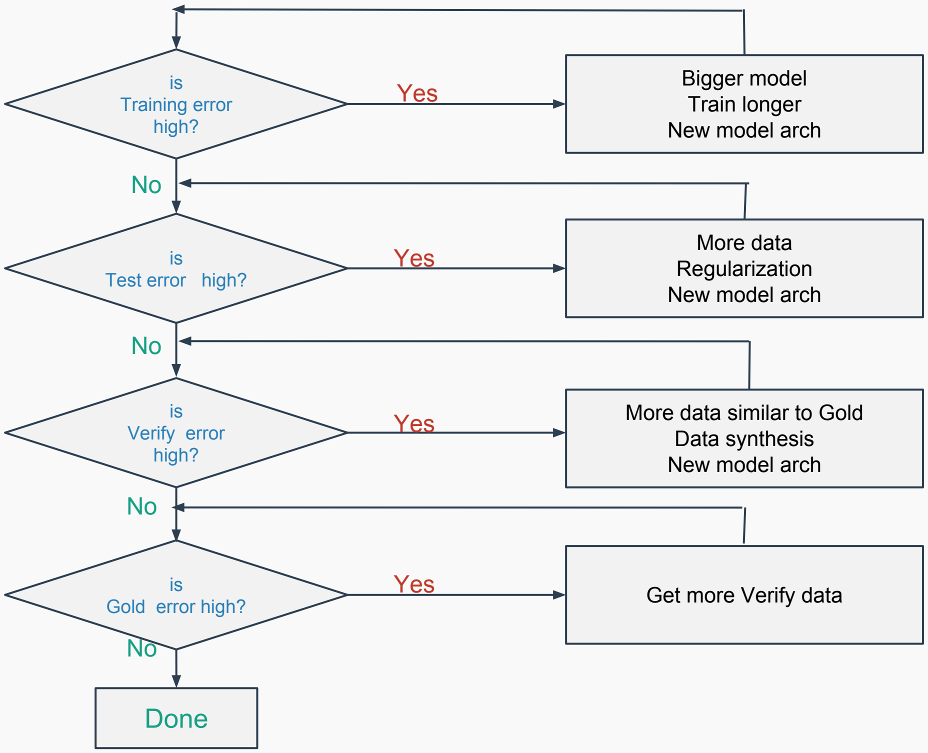
Anti-pattern 1 : Garbage in - Garbage out

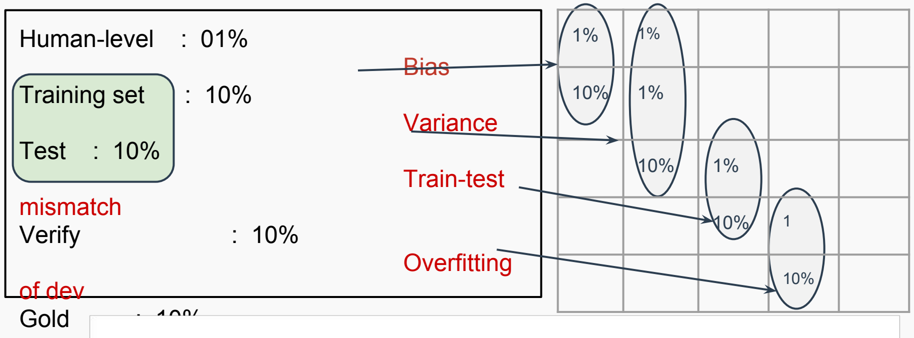
- Look at your data
  - Kosher: Nothing off in Input and Output
  - Do a simple walk through each stage
- Can you overfit your model (98-99%+ training)
- Visualize label/output distribution: Is it nearly equal, if not then handling uneven classes
- Improving Beyond (large data issues)
  - Tools to effectively view large quantity of data
  - Look at critical regions & Inspect your model: CAM/LIME
  - Is it really confusing or learning unrelated patterns (Bach vs Mozart)
- Synthesize or Augment or Simulate
  - GANs
  - Simulate

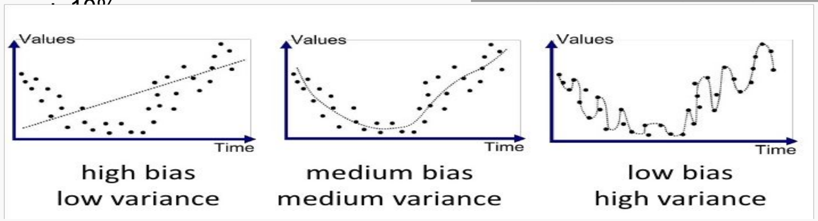
GAN Gen.

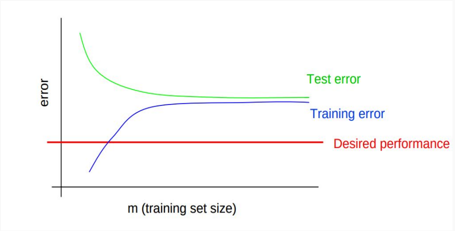
Driverless: Simulator

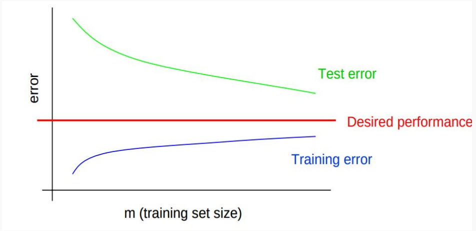
Disciplined ML/DL Training











Based on Andrew Ng ‘2013

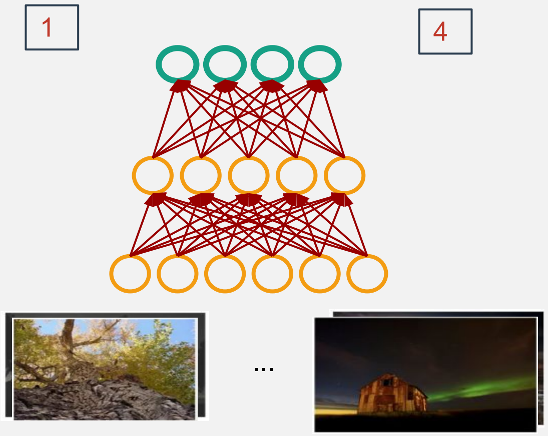
Anti-pattern 2 : Metrics (good, bad, ugly)

- Incorrect Metrics
- Bad Loss Function
- Non uniform dist of labels

Loss & Metrics

Aesthetic Scoring Problem

Training

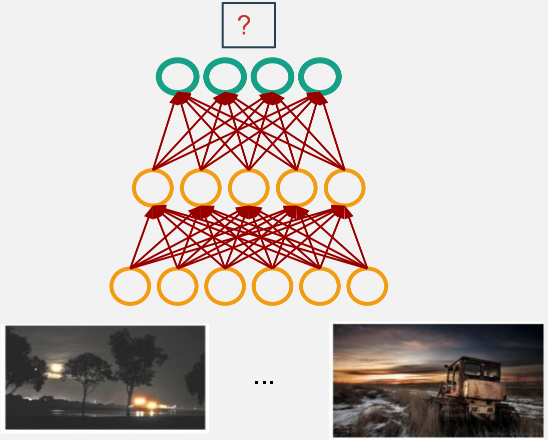


Loss: To optimize your model

Actual	4:Wow	4:Wow	4:Wow	4:Wow
Pred	1: Bad	2: Nice	3:Good	4:Wow
Loss A	1	1	1	0
Loss B	3	2	1	0
Loss C	9	4	1	0

Metric: To judge performance

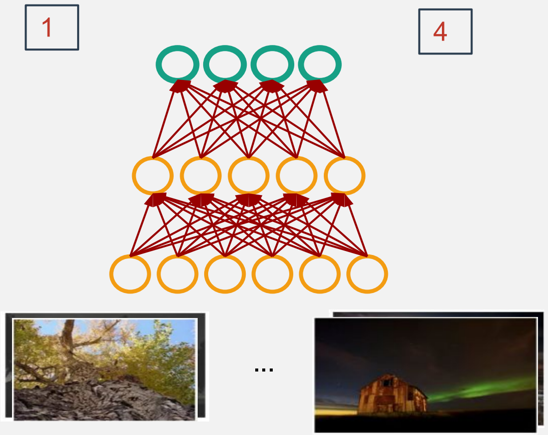
Testing



Loss & Metrics

Aesthetic Scoring Problem

Training

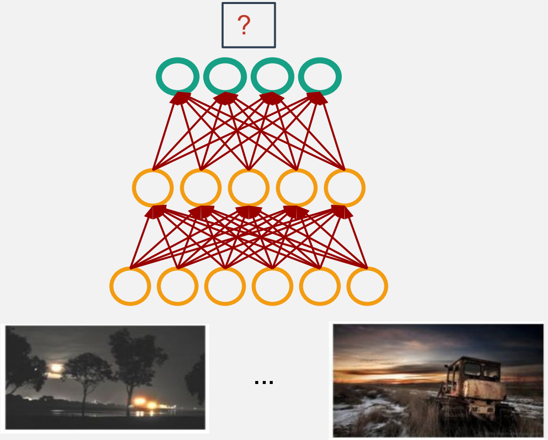


Loss: To optimize your model

Actual	4 (Cat)	4 (Cat)	4 (Cat)	4 (Cat)
Pred	1 (Dog)	2 (Pig)	3 (Man)	4 (Cat)
Loss A	1	1	1	0
Loss B	3	2	1	0
Loss C	9	4	1	0

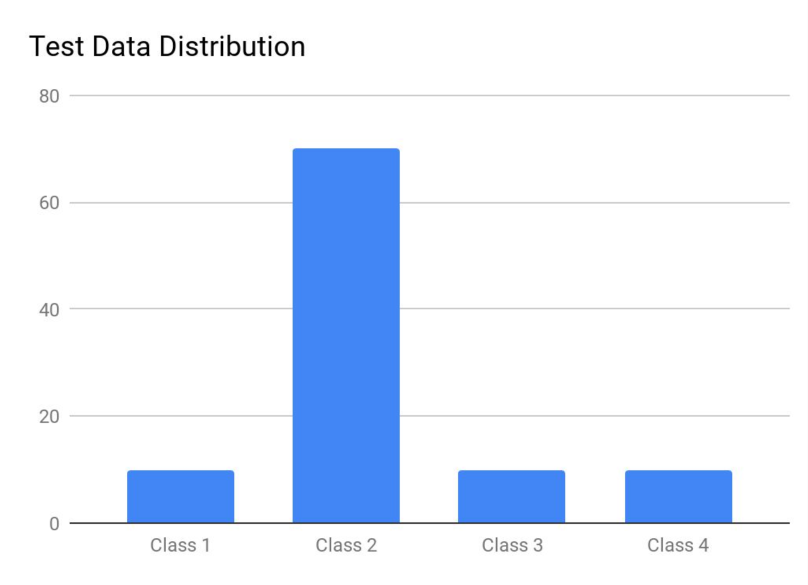
Metric: To judge performance

Testing



Anti-pattern 2 : Metrics (good, bad, ugly)

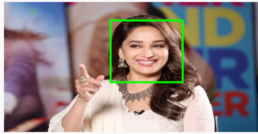
- Incorrect Metrics
- Bad Loss Function
- Bad distribution of data



Anti-pattern 2 : Metrics (good, bad, ugly)

- Incorrect Metrics (Harsh's Talk)
- Bad Loss Function
- Bad distribution of data
- No Context: Face Recog of 98 %
  - On benchmark + already detected
  - Detection & Localization vs Recog vs Verification
- Under very different conditions
- Incorrect Maths
- Not accounting for info leaks

Detection & Loc
Is there & Where are faces

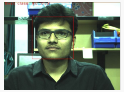


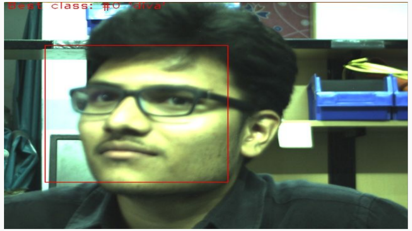
Authentication/Verify
Is she Madhuri Dixit ?

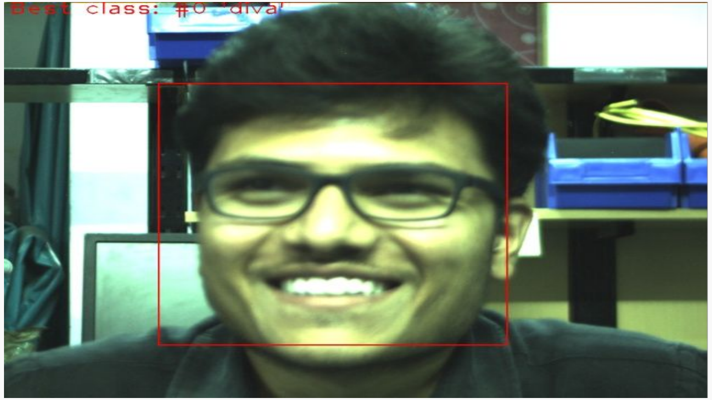
Recognition
Who is this ?

Anti-pattern 2 : Metrics (good, bad, ugly)

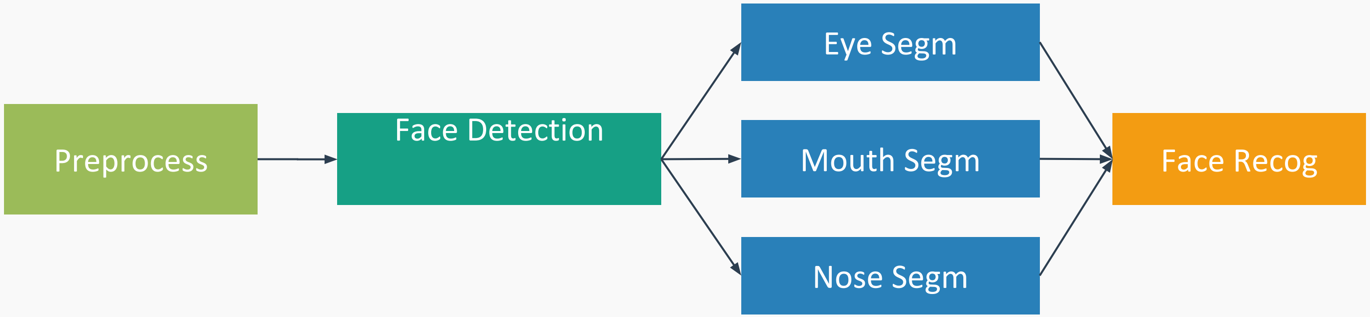
- Incorrect Metrics
- Bad Loss Function
- Bad distribution of data
- No Context: Face Recog of 98 %
  - On benchmark + already detected
  - Detection & Localization vs Recog vs Verification
- Under very different conditions
- Incorrect Maths: Loss/Metrics
- Not accounting for info leaks





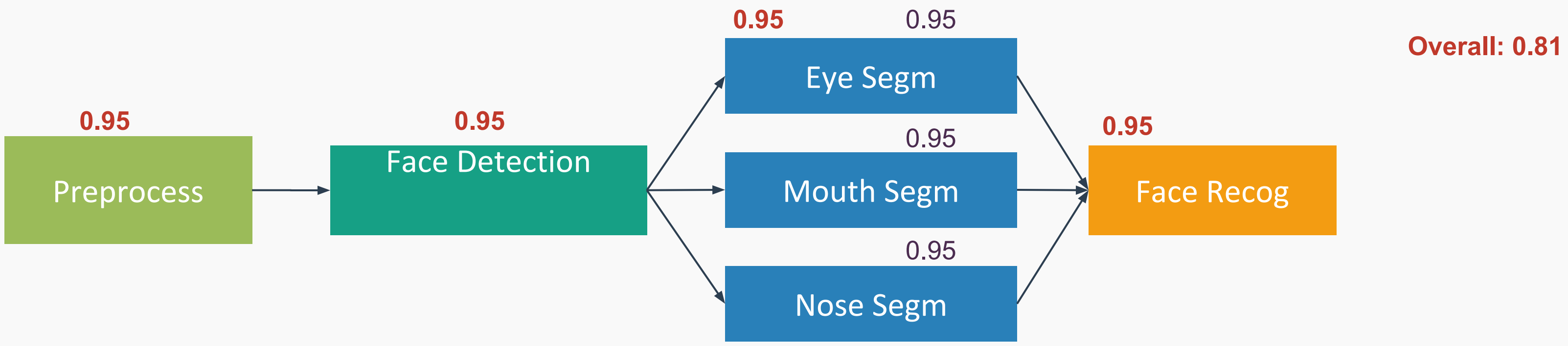


Anti-pattern 3 : Divide and Conquer



- Better Interpretability & Easier to debug
- Easier to improve
- Distributed Development / Dedicated Personal

Anti-pattern 3 : Divide and Conquer

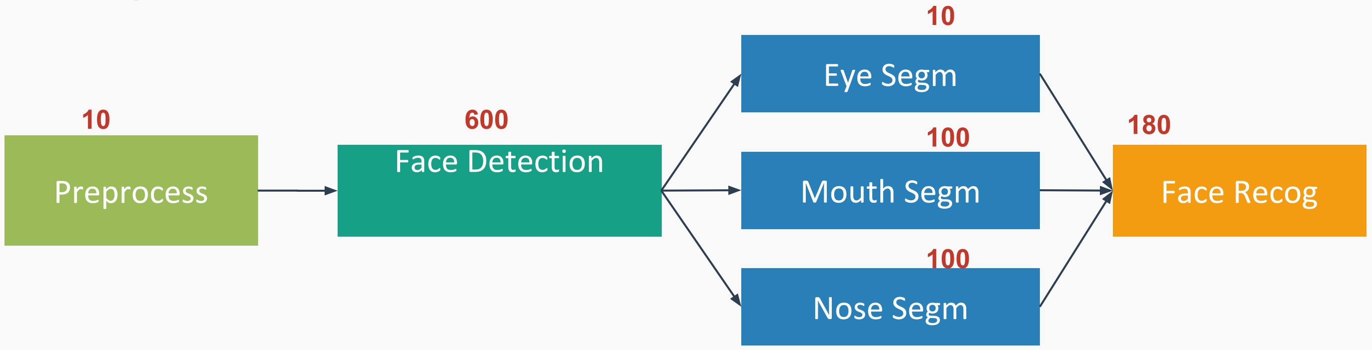


0.95 * 0.95 * min (0.95, 0.90, 0.95) * 0.95 = 0.81

- Error gets accumulated at each stage
- Not independent: Error cascades

Anti-pattern 3 : Divide and Conquer

Total : 10K Images



- Error gets accumulated at each stage
- Not independent: Error cascades
- Not independent: Improvement Cascades Needed (War Story)
- Happens in DL too: Two stage detector, CRF on top, CNN-RNN etc

Anti-pattern 4 : General-Enough vs Over-General

- Sales: More general it is, easier to sell
- Sales: high **accuracy** & **fast (30fps)** solution
- Engg: accurate & faster:

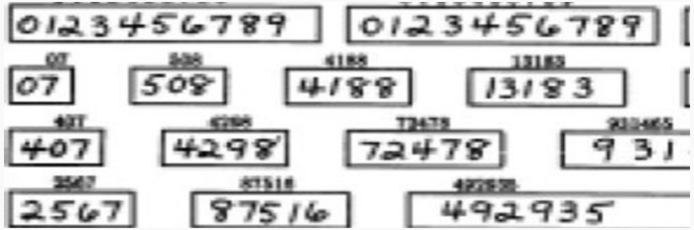
    Expensive to develop such a system (many man months)

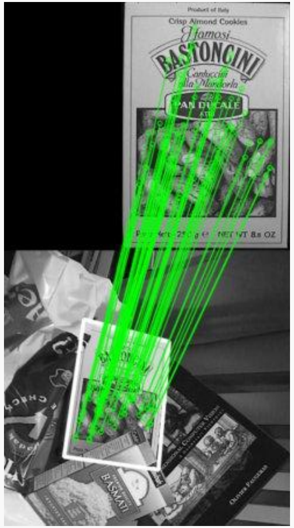
Anti-pattern 4 : General-Enough vs Over-General

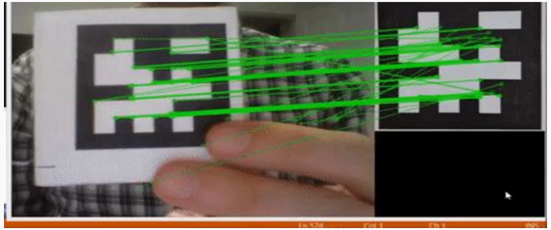
- **Computer Vision**: Types of variations
  - Scale, Rotation & 3D Rotation, Translation
  - Intra-Class variance: smaller better
  - Inter-Class variance: larger better
  - Lighting: Low light vs Specular Reflection etc
  - Occlusion

Anti-pattern 4 : General-Enough vs Over-General

- Sales: More general it , easier to sell
- Sales: high **accuracy** & **fast (30fps)** solution
- Engg: accurate & faster
- War Story: Form Reading Page Alignment
  - Alignment: 0-360 Degrees
  - 4-5 Mo: matched marker-descriptor
  - Accuracy: 98.8+%







Anti-pattern 4 : General-Enough vs Over-General

- Sales: More general it (seems), easier to sell
- Sales: high **accuracy** & **fast (30fps)** solution (for demos)
- Engg: accurate & faster
- War Story: Form Reading Page Alignment
  - Alignment: 0-360 Degrees
  - 4-5 Mo: matched marker-descriptor
  - Accuracy: 98.8+%
  - Real world scanner: 99.99%+
- Clear understanding of product use-case
- Generality is expensive, choose wisely (Engg Comm)

Anti-pattern 5 : Testing, Production etc

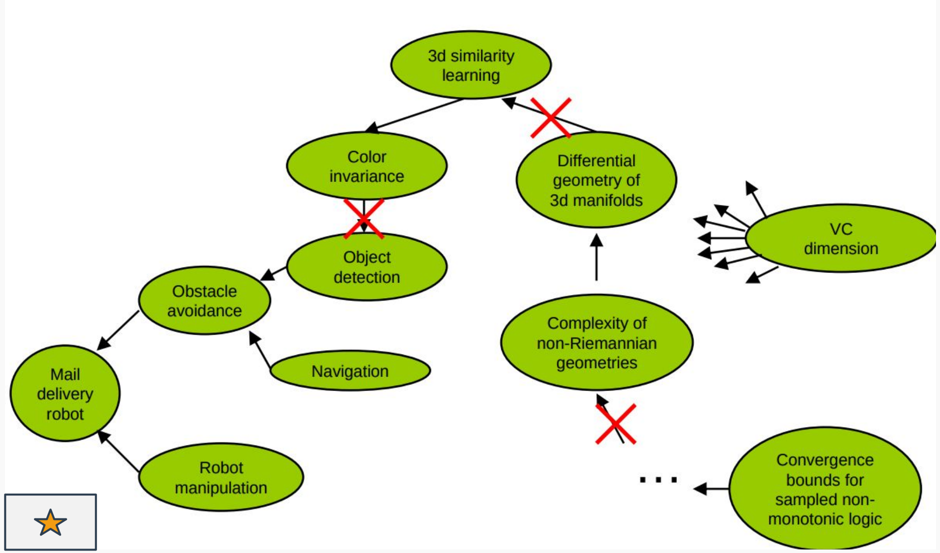
- Managers: Understand what problem team is really solving
- Managers: Engage in deeper conversations, allow it ok for engineers to **not** know/understand

  None of us know why these really works, though we do have some intuitions
- Site-Testing: Understand users creating your data
  - UI/UX causing bad data (words like intermediate, exceptional)
  - Children Binging behaviours (measures to reduce)
  - Good UI/UX to create quality data (to remind users/annotators)
- **Prod != Prototype**
- **Human in the loop**

Anti-pattern 7 : Over-theorize

Build and Iterate;

Ph.D's and Math inclined more prone to this (self-confession :)



Based on Andrew Ng '2007 and Papadimitriou, 1995

Takeaways

- **Managers et al**
    - **Not Magic Sparkle:** Systematic, Disciplined development, don't over trivialize,
    - Importance of **Clean Data** & **Representative Data**
    - **Trust your engineers** instincts but ensure they are solving the **right tight problem**
    - Understand **Prototype != Production**
    - Balance **General Enough vs Over General** (Over greedy is bad)
    - **Testing:** Importance of **UI/UX**, test often/test early, engineers must see what users does
    - **Man in the loop**: At-least Initially, faster to iterate
    - **Don't over-theorize**

Takeaways

- **DL Engineers et al**
  - **Garbage in - Garbage Out:** View your data, Systematic Debugging, Build Tools, Simulate/Augment
  - **Disciplined ML:** Different data sets
  - **Good, Bad, Ugly Loss and Metrics:** Context of publ. results, correct dist?, same conditions, Info leaks
  - **Divide and Conquer**: Interpretability and Error/Improvement Cascades
  - Keep **up-to date with tools**: spacy, keras, etc...

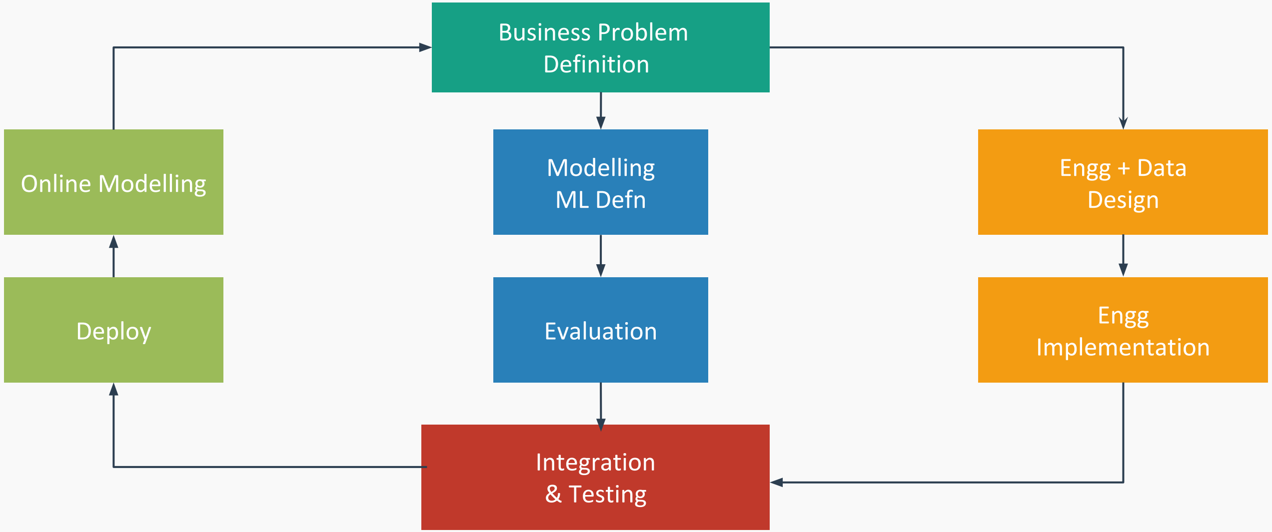
Thanks!

AutoInfer Technologies
14, 12th Cross Road, Vasanth Nagar
Bangalore 560052, India
sales@autoinfer.com

Soliton Technologies
683, 15th Cross Road, 2nd Phase, J. P. Nagar,
Bangalore 560078, India
vision@solitontech.com

Extra Slides

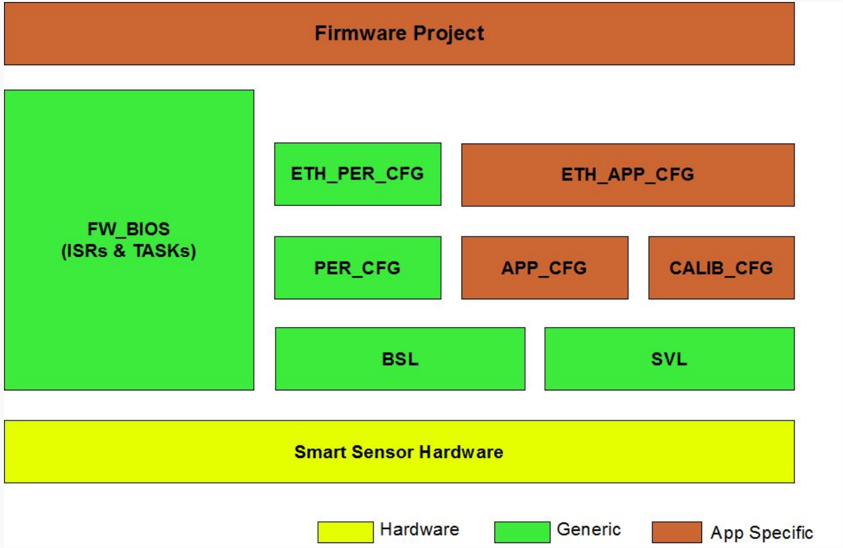
ML System Life Cycle



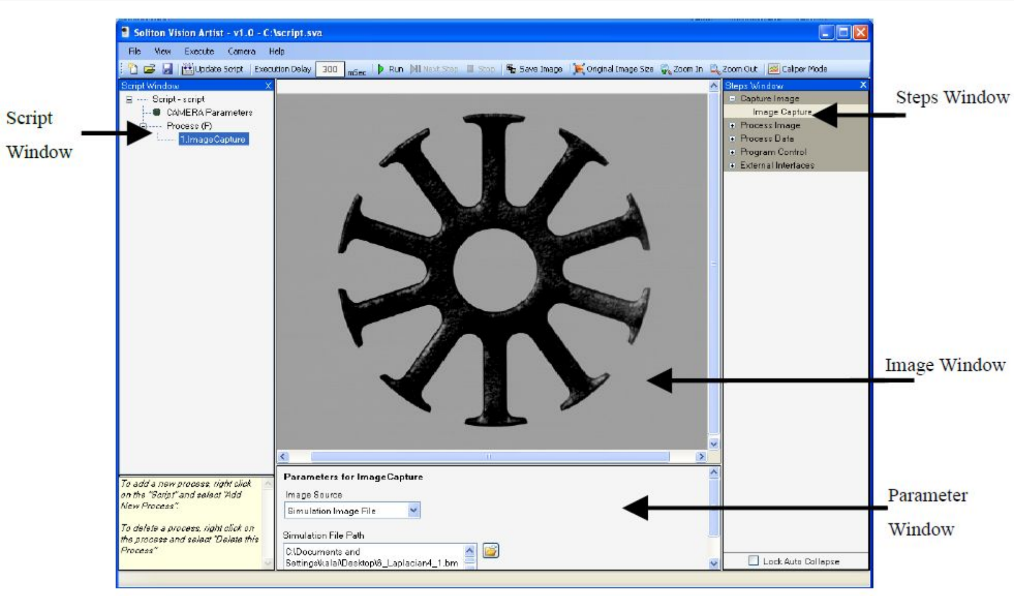
Takeaways

- **No Magic Sparkle:** Systematic, Disciplined development,
- **Defining Business Problem**
  - Assumptions can you make: Day/Night, 10K Objects
  - Cut Slack but solve only what is needed
- **Defining ML Problem**
  - Dividing into sub-problems (improves interpretability)
  - Shannon’s Successful Researcher (Error Propagation)
- **Modelling**
  - **Disciplined ML**: Dev, Valid, Test (datasets**)**
  - **Systems Thinking**: Handling
  - **Metrics**: Good Metrics, Bad Metrics, Ugly Metrics & What to optimize (segm eg)
- Understand Limitation, Incorporate Rich Data, Iterate with real data as soon as possible
  - Setting Expectations

Soliton NEO Architecture

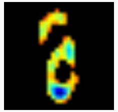


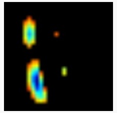
Soliton Vision Artist



Visualization - Class Activation Maps

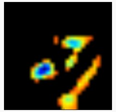

3 as 2


2

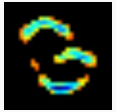

3

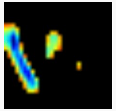

4 as 6

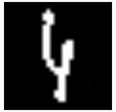

6

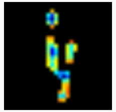

4

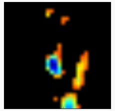

3 as 0

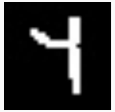

0

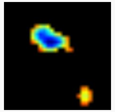

3

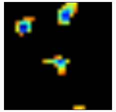

4 as 6


6

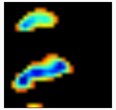

4

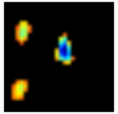

7 as 4


4


7

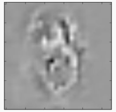

3 as 7


7


3

Visualization - Class Activation Maps

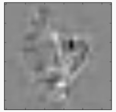
3 as 2



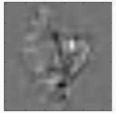
2

3

4 as 6

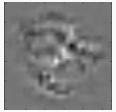


6

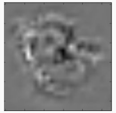


4

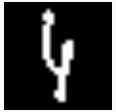
3 as 0



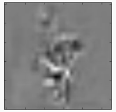
0



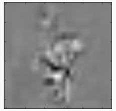
3



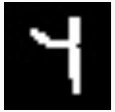
4 as 6



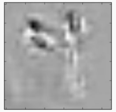
6



4



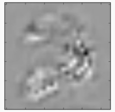
7 as 4



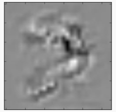
4

7

3 as 7



7



3

Visualization - Class Activation Maps