CNN - Convolutional Neural Networks

Convolutional neural networks, or CNNs, are specialised architectures that work particularly well with visual data, i.e., images and videos. A CNN is a multistack layer of simple modules: convolution layers, activation functions, Max pooling, Dense and output predictions.

Common Interview Questions

1. What are common vision applications in Retail, Healthcare, Energy?
2. What are the different stages of end-to-end pipeline of vision use case?
3. What are challenges / myths building vision applications ?
4. What is difference between instance and semantic segmentation ?
5. What are key components of U-net ?

CNNs: Industrial Applications

- Retail
- Fashion
- Energy
- Surveillance
- Manufacturing

Vision Industrial Applications

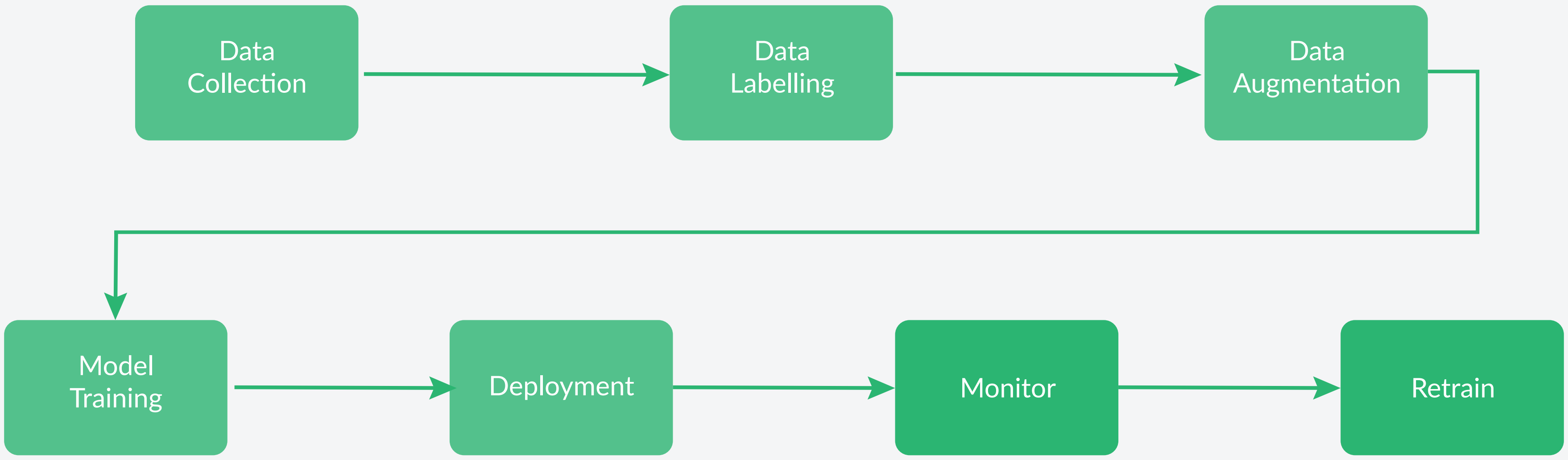
Computer vision use case across different industries

- **Fashion** - Automated Cloth attributes, Celebrity looks, Sequence / Scene / Image tagging / Aesthetics extraction
- **Retail** - Product detection / Shrink management / People counting based on store CCTV images
- **Beauty** - Virtual hair color, hair styler, accessory try-on based on segmentation
- **Agriculture** - Plant identification, Crop disease identification
- **Insurance** - Damage assessment, Defect Detection of vehicle damage
- **Vision in healthcare** - Xray, MRI, Scan images based anomaly/ disease detection
- **Vision for BPO Automation** - Automated invoice processing
- **Vision for Monitoring / Inspection** - Quality Assessment
- **Vision for Safety** - Industrial safety
- **Vision for Fruit Freshness** - Automated pricing on quality

Computer Vision End to End System

Building Vision Solution requires workflow of components

- **Data Collection** - Collect data, and samples based on the field of view
- **Data Labelling** - Label Data for training the model
- **Data Augmentation** - Rotate, Sharpen, Add noise
- **Training / Optimization** - Build Models based on CNN / Transfer Learning
- **Deploy** - Deploy a model as Flask API / Rest API endpoint
- Monitor - Monitor the model, Perform Field Testing
- **Retrain** - Based on Field Test results, Retrain the model



Segmentation

Image segmentation can be thought of a classification task on the pixel level,

Types of Segmentation

- **Semantic segmentation** classifies every pixel in a given image into a class.
- **Instance segmentation** deals with detecting instances of objects and demarcating boundaries
- Methods can be both R-CNN and FCN (Fully Convolutional Networks) driven.

Architectures

U-NET

This is a convolutional neural network originally created for segmenting biomedical images. The architecture is made of two parts:

- Left part: the contracting path, which captures context. Made of two three-by-three convolutions, which are followed by a rectified linear unit and a two-by-two max-pooling computation for downsampling.
- Right part, the expansive path, helps in precise localization.

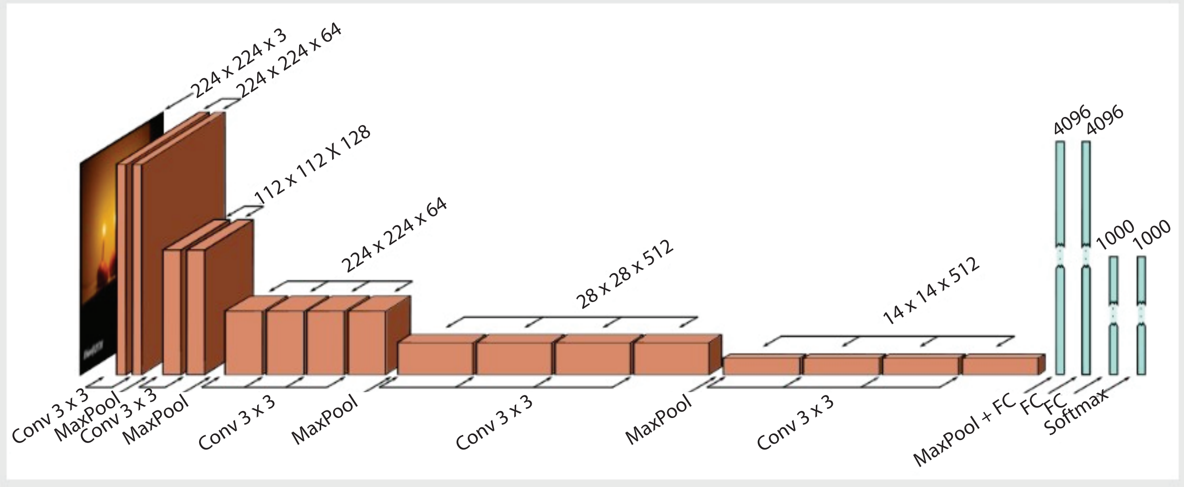
Mask R-CNN

Objects are classified and localized through a bounding box and semantic segmentation, which classifies each individual pixel into a set of categories.

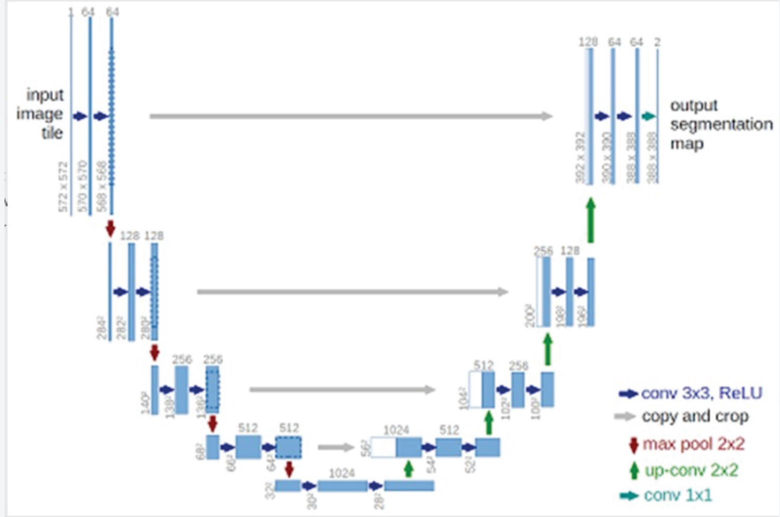
- All regions of interest get a segmentation mask and a class label and bounding boxes are produced as the final output.
- This architecture is an extension of Faster R-CNN, which is composed of a deep convolutional network proposing regions and a detector utilizing regions

Activation Function Insights

VGGNet - VGG19 is a convolutional neural network model developed by the Visual Geometry Group at Oxford University. It is a 19-layer network, consisting of 16 convolutional layers and 3 fully connected layers. VGG19 is a powerful model for image recognition and classification tasks, achieving state-of-the-art results on many benchmark datasets.



Unet - Unet model has horizontal connections between the down sampling and up sampling layers. These connections are known as skip connectionsand they allow the model to pass information from the down sampling layers to the up-sampling layers. This helps the model learn more complex features and improves its performance

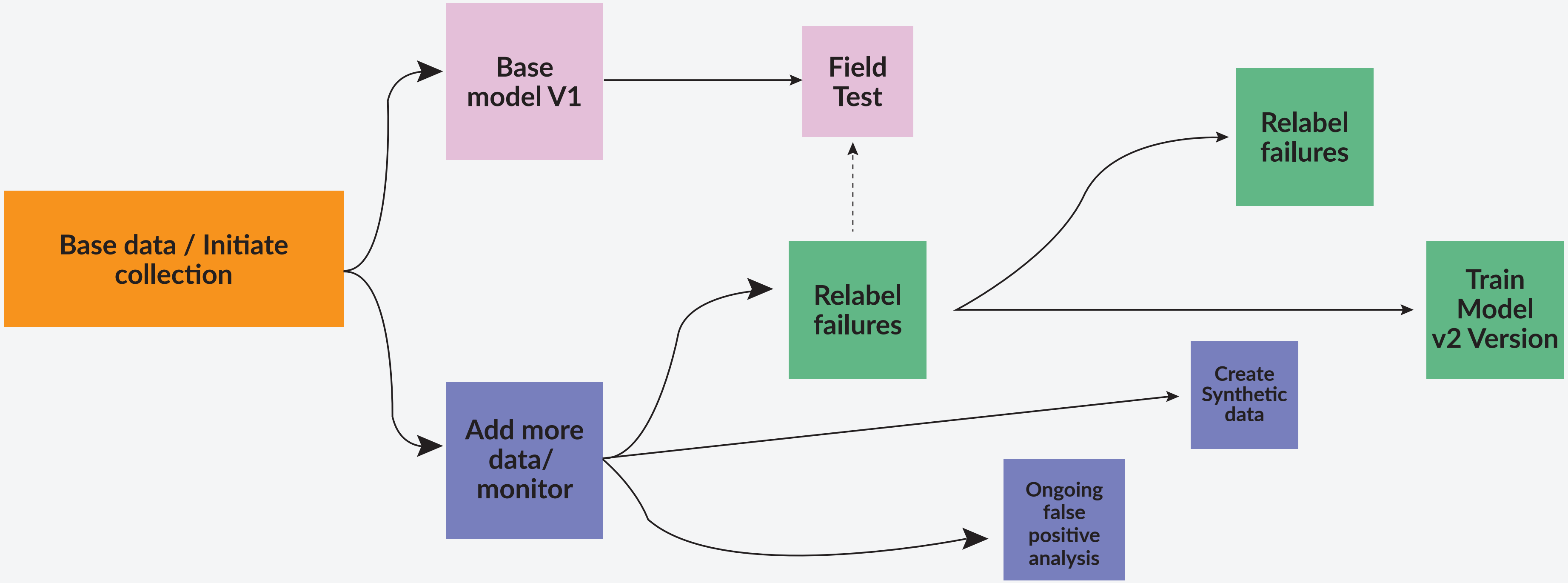


Architecture Comparisons

	Alexnet	VGGNet	GoogleNet	Resnet
Overview	AlexNet was developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012 . It was also the first to use the ReLU activation function	VGGNet is a convolutional neural network architecture proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" in 2014.	GoogleNet was first introduced in 2014 by Google researchers Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich.	ResNet is a deep learning architecture developed by Microsoft Research in 2015.
Number of Layers	AlexNet consists of 8 layers, 5 convolutional layers and 3 fully-connected layers.	Number of Layers: VGGNet consists of 16 layers, 13 convolutional layers, and 3 fully connected layers.	GoogleNet consists of 22 layers, including 9 Inception modules.	It consists of 152 layers and is one of the most accurate image classification models available today.
Accuracy	AlexNet achieved a top-5 accuracy of 80.2% on the ImageNet dataset.	VGGNet has achieved very good accuracy on the ImageNet dataset, with a top-5 error rate of 7.3%.	GoogleNet has achieved an accuracy of up to 93.3% on the ImageNet dataset.	ResNet has achieved an accuracy of up to 95.8% on the ImageNet dataset.
Pros	AlexNet also introduced the concept of dropout regularization, which is now a standard technique for preventing overfitting in deep learning.	• It uses a very small convolutional filter size of 3x3, which helps reduce the amount of parameters and computation, while still being able to capture complex features.	• It uses fewer parameters than other deep learning architectures, making it faster and more efficient. • It is capable of recognizing objects in images with high accuracy.	• It uses fewer parameters than other deep learning architectures, making it faster and more efficient. • It is capable of recognizing objects in images with high accuracy.
Cons	It also does not perform well on small datasets			• ResNet is computationally expensive and requires a lot of memory to run.

Perception vs Reality

- **Perception 1** - We need millions of images to build models, We need similar huge datasets.
- **Reality** - We do not need to wait for perfect data, Data collection, or Synthetic data creation everything is a going process, Start Small, Keep Iterating, Shipping / Innovating
- **Perception 2**- Data collection is effortless, It can be done by google search / kaggle
- **Reality** - The real world and the kaggle dataset are miles apart. Real-world challenges are dependent on light/angle/hardware used. Buying data is even more expensive :). #Data cost is more #costly than model training time
- **Perception 3** - I need the start of the art model with 99% accuracy / Can we get a performance like the state of the art?
- **Reality** - We need to be realistic with the data we have, and an incremental model that we can develop.
- **Challenge / Perception 4** - Model development is a one-time effort. Collect / Build / Deploy / Move on
- **Reality** - Base model / retrain / field test and next version is incremental effort. ML is an iterative incremental effort. It has a set of parallel ongoing efforts like below
  When the customer wants state of art but has no strategy on how they need to incrementally build upon becomes an effective challenge to provide the vision/clarity.


Base data / Initiate collection
Base model V1
Field Test
Relabel failures
Relabel failures
Train Model v2 Version
Add more data/ monitor
Create Synthetic data
Ongoing false positive analysis