DNN Training Data: How to Know What You Need and How to Get It

Abhishek Sharma
Edge AI and Vision Alliance

Tech Mahindra

What we would cover

- Data consideration for learnability – ERM & PAC
- Problem of paucity
  - Occam's Razor approach for data – Classical vs Deep Learning , Transfer Learning , Open Data Sets
  - Data augmentation (image /video , text , speech)
  - GANs for Data augmentation
- Problem of plenty – Constrained Resource Optimizations
- Case Studies
- Conclusion
- Future Work

ERM & PAC – Training Data

- Empirical Risk Minimization
  - Risk - We don't know the true *distribution* of data on which an algorithm will work in its *inference* phase
  - Empirical - We measure performance of an algorithm on a known set of training data in the *training* phase
  - Minimization – Choose the hypothesis which minimizes the risk.
    - Understand the problem context and device solution as applicable (classical or deep learning)
    - DRY (Don't Repeat Yourself) Principal – Code & Data (transfer learning / open-source data sets)
    - Perpetual Learning – Account for data volumes and appropriate algorithms (for example Bayesian algorithms work well for low data volume) , model life cycle is also critical in terms of hot swapping models as more data is collected and as other model is giving better performance .

ERM & PAC – Training Data

- Probably Approximately Correct (PAC) learnability – In simple terms it states, for a given problem, if we consider multiple possible classifiers (hypothesis is set of classifiers), to select the one with lower error and higher probability of correctness, *we need more data to distinguish between them.*

$$m \geq \frac{1}{\epsilon}(ln|H| + ln\frac{1}{\delta})$$

- *m* is the sample size which depends on the set of hypothesis classes H
- *h*, a subset of H, is *approximately* correct if it is bounded by the error over the distribution (D) of data in *m* by some *epsilon* ε, i.e., Error (D) < ε
- *Correct* is the probability (1 – δ) that the learning algorithm will output such a classifier .

4

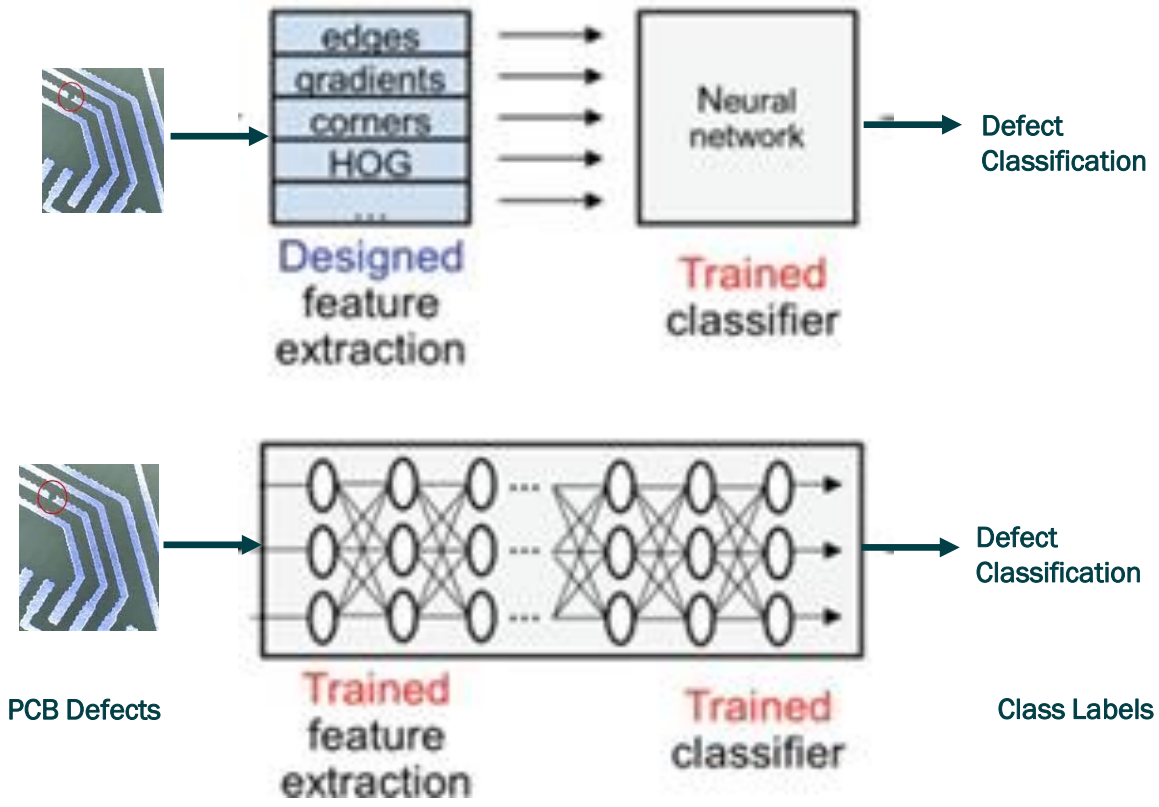
Occam's Razor Principal

- Occam's Razor Principal - In comparing two models that provide similar predictions or descriptions of reality, we should select for the one which is less complex.

- Finding a deep learning model to perform well is an exciting feat. But, might there be other -- less complex -- models that perform just as well for your application.
- We advocate finding a strategy for deep learning data paucity challenge, which is:
  - Choosing the right algorithm – classical vs deep learning
  - Leverage transfer learning
  - Open data sets

Classical vs Deep Learning

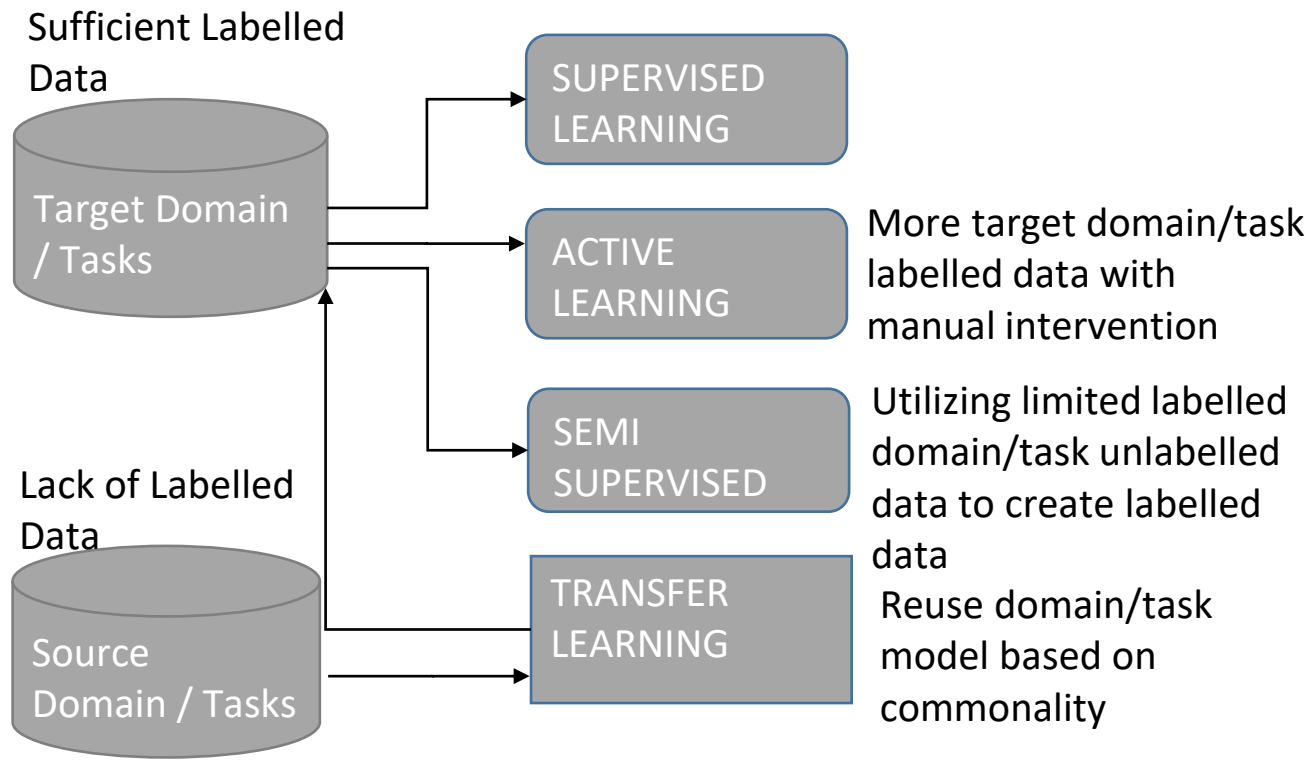
	Classical Algorithms	Deep Learning Algorithms
Requirements		
High Quantity of Labelled Data Set	NO	YES
Manual Feature Extraction	YES	NO
Deployment Ease on Microprocessor	YES	NO
High Accuracy	NO	YES
Blackbox Models	NO	YES
Compute Intensive training.	NO	YES



Choice based on business problem context

Transfer Learning

- Transfer learning – For contexts where very little labelled data is available, we must leverage transfer learning.
- Handles data sparsity well in heterogeneous (Feature Space and Label Spaces are different) and homogeneous (FS and LS are same) contexts (e.g., using computer vision for defect identification in automotive extended to aerospace)



Transfer Learning - Approaches

Approaches of Transferring Knowledge

- Instance-based – Implicit assumption is overlapping of feature space between source and target domains, though due to the domain difference, source domain data cannot be used directly but part of them can be reused for the target domain after reweighting or resampling.
- Feature-based – Corresponds to the subspace spanned by the features in the source and target domains, good feature representation for both the source domain & target domain enables projecting data onto the new representation in target domain for precise classification
- Model-based – Implicit assumption is models in source domain have captured general structure which can be applied to the target domain, both domains share some parameters/hyperparameters for learning
- Model relation – Corresponds to rules specifying the relations between the entities in the source domain, unlike above approaches data in the source domain and the target domain are not required to be independent and identically distributed.

Transfer Learning – Methodology

Methodologies which can be deployed for DNNs

- Keep lower layers fixed & optimize parameters for higher layers.
- Remove the last N layers of a large network.

Open Data Sets

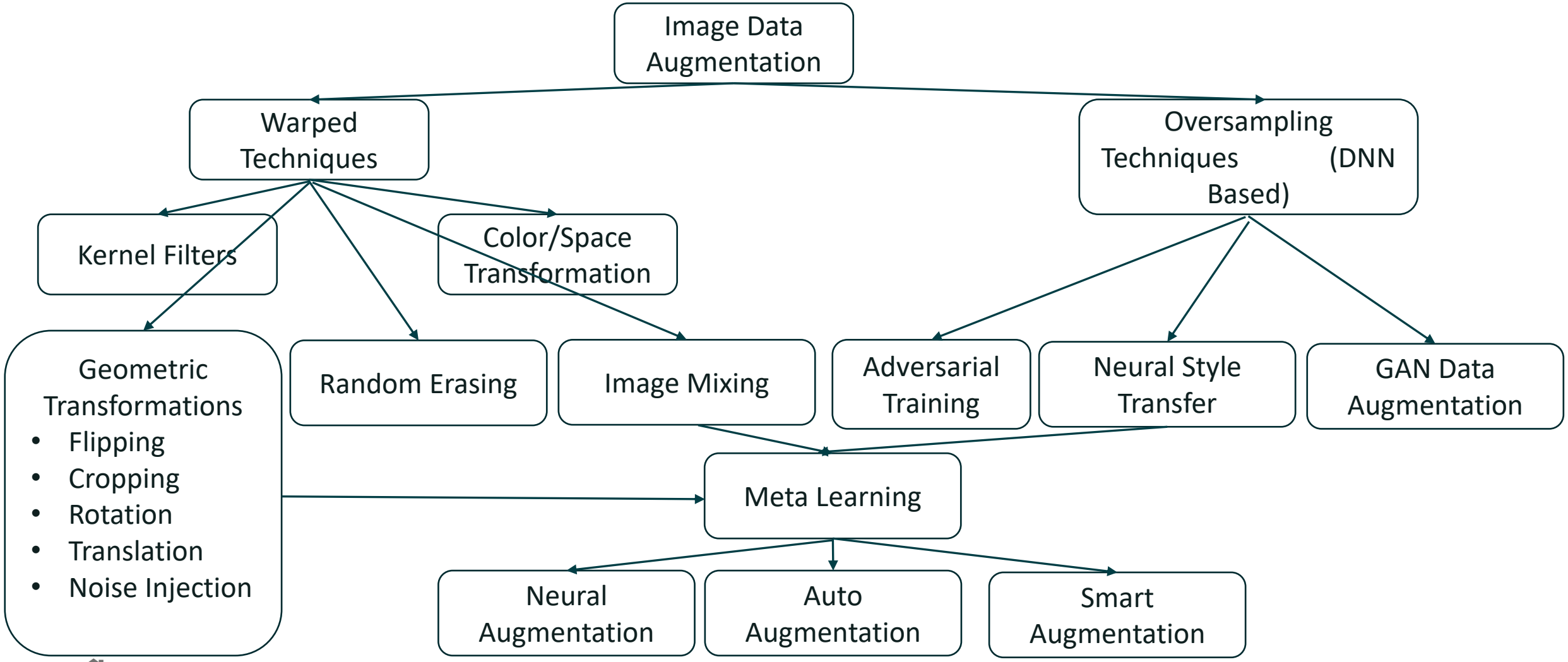
- Open data sets
  - https://datasetsearch.research.google.com/
  - https://www.kaggle.com/datasets
  - https://github.com/awesomedata/awesome-public-datasets
  - others

LEVERAGE PUBLICLY AVAILABLE DATA

Data Problem of Paucity - Simple but Not Simpler

- Data augmentation
  - Image/video data augmentation
  - Text data augmentation
  - Audio data augmentation

Data Problem of Paucity - Image/Video Data Augmentation



Considerations for Data Augmentation

- Invariance - Translation, Viewpoint, Size or Illumination

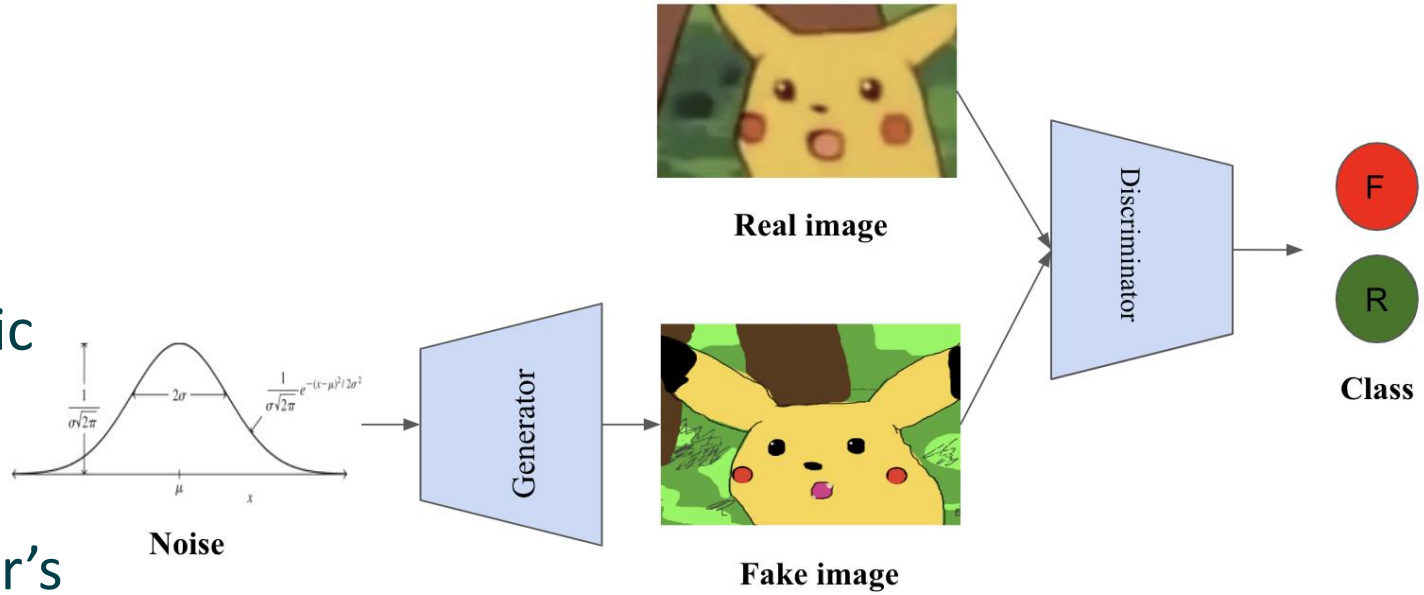
Translation and rotation invariance: DNN must learn being invariant to rotation.
How: Images can be flipped (horizontally/vertically) , changes to pixels and added back to training dataset.

Scale invariance: CNNs are unable to recognize objects at different scales , to learn DNN must be invariant to scales
How: Augment the training set with random crops of the input images. These crops may be sub-sampled version of training images or up-sample to the original height and width of the input image.

Color perturbation: DNN must learn accounting for illumination.
How: Training set images could be augmented by perturbing the color values of the input image directly.

GANs for Data Augmentation

- Unsupervised generative models which implicitly learn an underlying distribution.
- Learning process is a minimax game between two networks
- Generator, which generates synthetic data given a random noise vector
- Discriminator, which discriminates between real data and the generator's synthetic data.



GANs for Data Augmentation

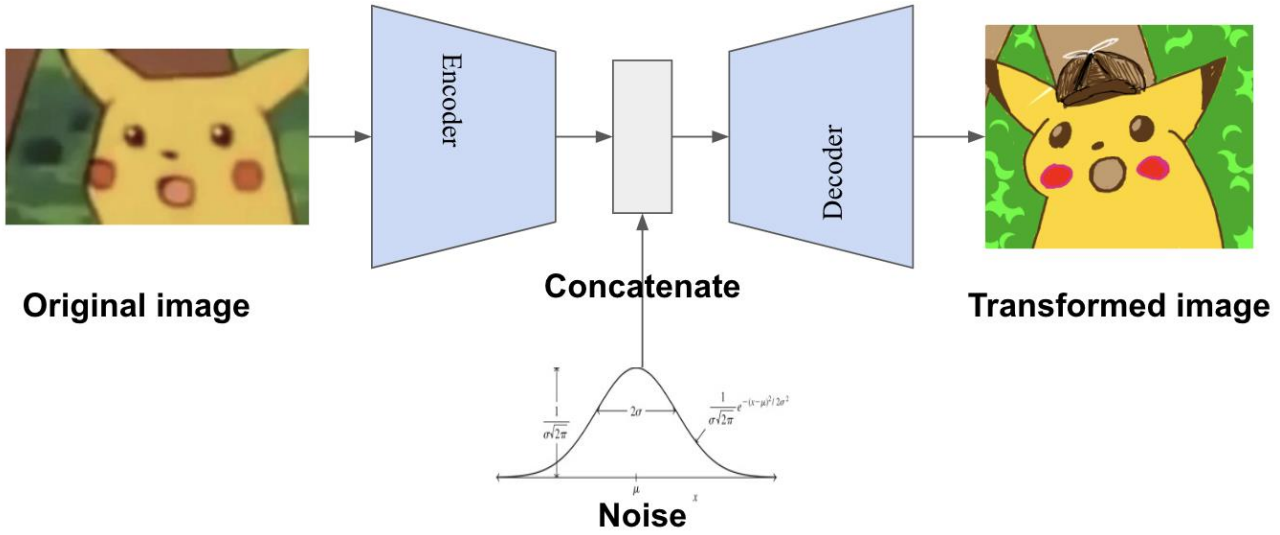
Applicable to transfer learning or few-shot learning use cases for data augmentation using GANs where we need to include class information in our GAN model.

Three types of conditional GANs where class information is fed to Generator :

- ACGAN (Auxiliary Classifier GAN)
  - Discriminator performs classification and discriminates between real and synthetic data
  - Loss function includes a binary cross-entropy term for classification to incentivize the generator to learn representative class samples in addition to learning to generate samples which are realistic overall.
  - Multitask learning generator and discriminator are “competing” on whether a generated image is real or fake, they are “cooperating” on classifying it correctly.

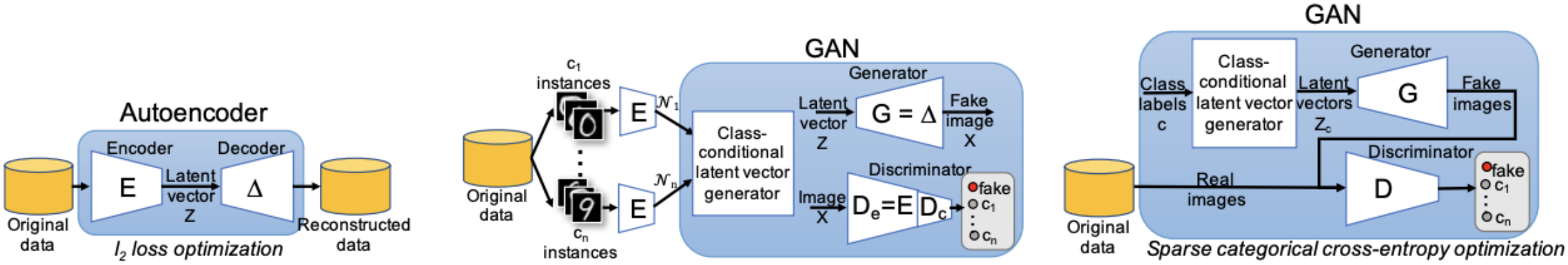
GANs for Data Augmentation

- DAGAN (Data Augmentation GAN), learns how to generate a synthetic image using a lower-dimensional representation of a real image.
- Rather than generator taking as input a class and noise vector, in the DAGAN framework, the generator is essentially an autoencoder
- Autoencoder takes an existing image, encodes it, adds noise, and decodes it.
- Decoder learns a large family of transformations for data augmentation.



GANs for Data Augmentation

- BAGAN (BAlancing GAN), an autoencoder is also used for the generator.
- Autoencoder is pre-trained to learn the distribution of the overall dataset.
- Multivariate normal distributions are fit to the encoded image of each class and samples can be taken from these multivariate normals and passed on to the resulting *conditional* latent vector to the generator.
- BAGAN gives full-fledged conditional generator, rather than transformations for existing data as in DAGAN
- DAGAN performs better than ACGAN in a few-shot context, because of the VAE's ability to learn the overall distribution before fitting normals to each class.



(a) Autoencoder training. (b) GAN initialization. (c) GAN training.

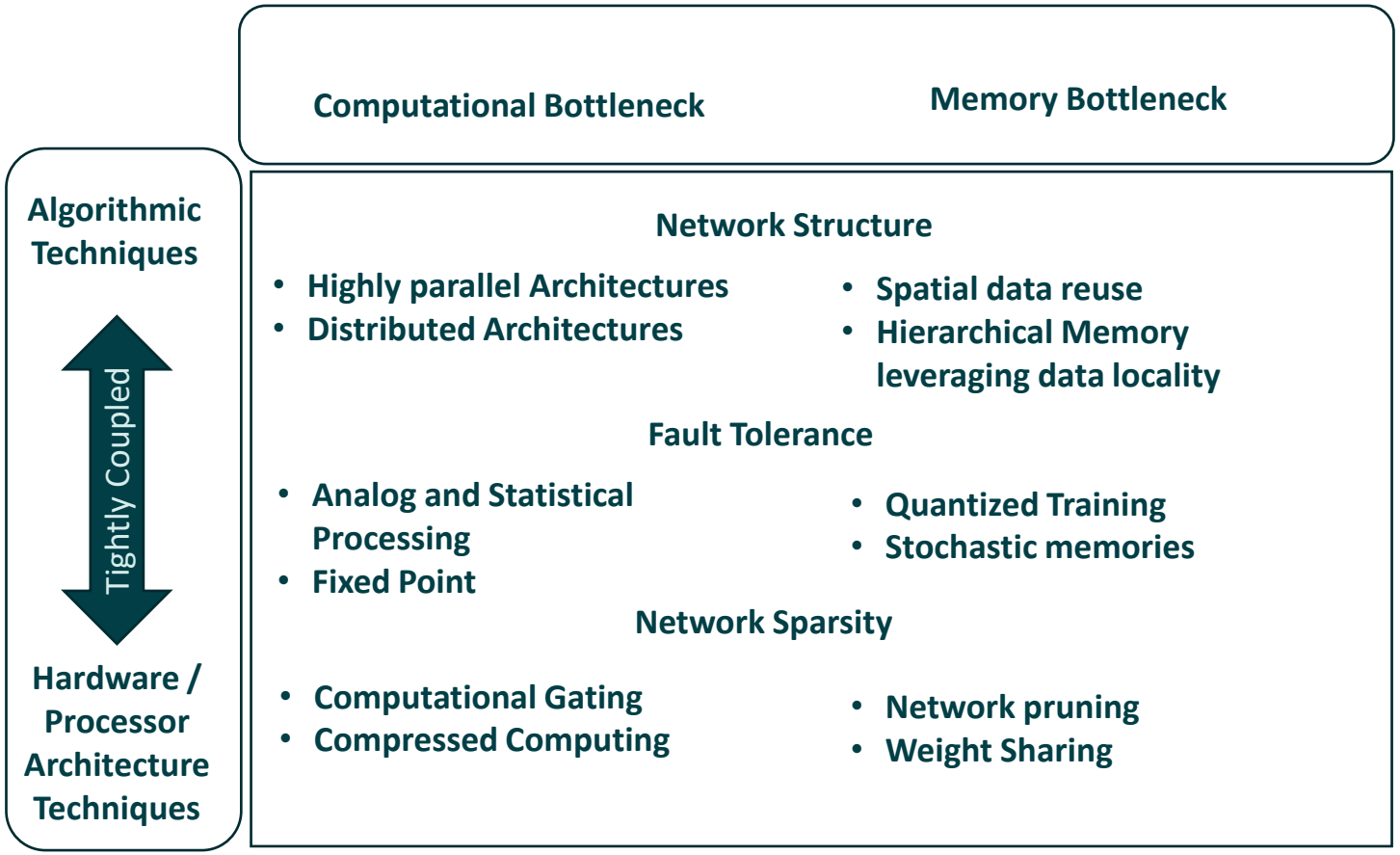
Delineation of DNN Data Augmentation in Text/Audio

2021 embedded VISION summit

AUDIO	TEXT	TIME SERIES
• NOISE INJECTION • TIME SHIFT • TIME STRETCHING • RANDOM CROPPING • PITCH SCALING • RANGE COMPRESSION • SIMPLE GAIN • EQUALIZATION **Libraries** • Nympy • Librosa • Audiomentations	• THESAURUS • CONTEXTURALIZED • BACK TRANSLATION • TEXT GENERATION LIBRARIES Libraries • NLP Albumentation • NLPAug	• CROPPING • WARPING • FLIPPNIG • JITTERING • TIME SHIFT • ADVERSARIAL NOISE • GANS Libraries • TSAug

Tech Mahindra

Problem of Plenty – Approaches



Algorithmic and Processor architecture techniques to optimize Neural Networks with Resource Bottlenecks

Problem of Plenty – Solution overview

- Data path optimization : Minimize data movement, maximize parallelism , maintain flexibility. Optimizing Multiply Accumulate Operations :
  - Input stationary: same input data is multiplied with several weights of different output channels of a layer
  - Weight stationary: every weight is fetched once and multiplied with many input values improves weight memory bandwidth
  - Output stationary : reloads new weights and inputs every single clock cycle, yet is able to accumulate the intermediate results across different clock cycles, to the benefit of the output memory bandwidth

Problem of Plenty – Solution overview

- Approximate computing :
  - Reducing precision from32b floating-point to lower fixed-point precision reduces computational energy, minimizes storage and data fetching cost network weights and intermediate results
  - Binary-weight version of ImageNet is only 2.9% less accurate (in top-1 measure) than the full-precision AlexNet
- Optimized Hierarchical Cascaded Processing :
  - Hierarchy starts with a simple binary classification that removes the most obvious negative samples like background images or acoustic noise stages.
  - Each stage, only a few layers of the network are executed, classifiers are run, until a classification with distinct probabilities is obtained.

Use Cases

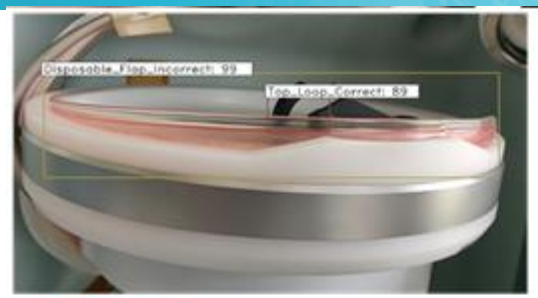
- Manufacturing
- Healthcare
- Semiconductor

Cognitive Inspection : Painted Class A Surface

- Business need : Visual inspection of class A surface of two wheeler
- Data challenge : 4 main categories of defects and no images /videos of data
- Solution : Set up of sample rig in test labs with cameras to take pictures with variations, illumination, etc., as close to production.

CV Enabled Medical Device Procedure Validation

- Business need : Visual inspection of maintenance of plasma extractor
- Data challenge : 8 main categories of defects and no images /videos of data
- Solution : Set up to utilize Tensorflow data augmentation for a quick turn around over 98% accuracy.

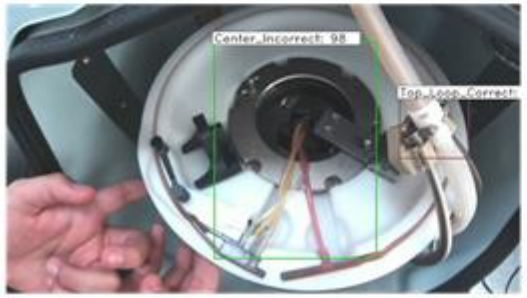


In Correct: Flap is protruding and not properly fit

Correct: Flap is properly fit

In Correct: Top loop is not in the groove

Correct: Top loop is in the groove

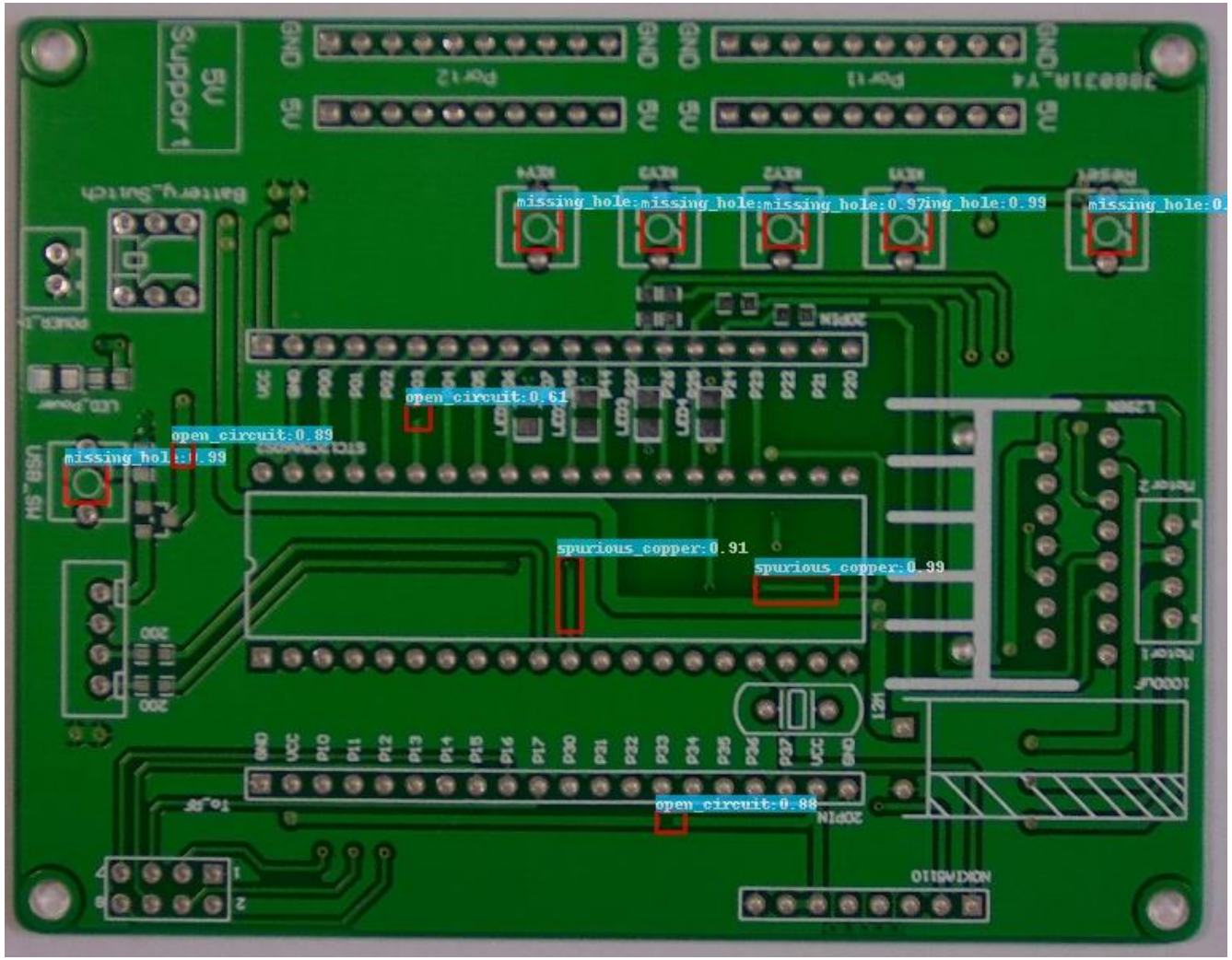


In Correct: Center wires are overlapping / crossing one over the other.

Correct: Center wires are not overlapping

AI Enabled PCB Defect Detection

- Business need : PCB quality checking is challenging due to the micro nature of defects and high rate of production
- Data challenge : Objective was to intelligently identify and report the location of the defects like missing hole, mouse bite, open circuit, short, spur & spurious copper on the given image of the PCB board. We had just 5-6 images per defect type which didn't suffice for an accurate solution
- Solution: Deep learning required 10X images where we utilized Keras-based data augmentation defect classification and detection with 97% accuracy.



PCB Defect detection in action

Conclusion

- Learnable Algorithms need data as a function of accuracy and error.
- Lack of data leads to overfitting of model
- To prevent overfitting we should take simplest approach possible i.e. Occam's Razor – Right Algorithm, Leveraging Transfer Learning , Open Data Sets
- Paucity of data leads to overfitting solutions approach for this could be in Dropouts, Batch Normalization
- Data Augmentation is one of ways to prevent overfitting which works on fundamental problem of data paucity by either warping or oversampling
- Warping maintains labels of images and applies transforms (geometric/color), neural style transfer , random erasing , adversarial training.
- Oversampling constitutes approaches which cater to synthetic data creation via feature space augmentation , mixing of images , GANs

Conclusion

- Hybrid approach with both warping and oversampling can be used.
- GANs – different variations like ACGAN, BAGAN, DAGAN can be leveraged contextually
- Problem of plenty we surveyed approaches from Hardware to CPU for constrained resource optimizations in DNN.

Data Paucity : Future Worrk

- “Less than one”-shot learning: useful in contexts where a classification algorithm is applied to a dataset with a large number of classes, especially if there are few, or no, examples available for some classes. Use soft labels to create new classes by partitioning the space between existing classes

  Specifically for DNN, Lo Shot Learning can be used if there are no examples available for some of the objects out of thousands of images/videos.

- Data Distillation : Is inspired from Network Distillation , typically model is fixed and knowledge of entire training dataset, which typically contains thousands to millions of images is encapsulated into a small number of synthetic training images.

References

Resources

https://arxiv.org/abs/1811.10959v3

https://arxiv.org/abs/2009.08449

Deep Learning Essentials by Wei Di, Anurag Bhardwaj, Jianing Wei`

https://towardsdatascience.com/binary-neural-networks-future-of-low-cost-neural-networks-bcc926888f3f

https://medium.com/abacus-ai/gans-for-data-augmentation-21a69de6c60b

https://awesomeopensource.com/projects/data-augmentation