Industry Applications, Career Path - Hands-on Image Classification and Text Generation

Premanand S

Assistant Professor
School of Electronics Engineering
Vellore Institute of Technology, Chennai Campus

premanand.s@vit.ac.in

March 28, 2025

Technology?

Why Technologies?

- **Solving Problems** – Technology addresses real-world challenges.
- **Improving Efficiency** – Faster, better, and automated processes.
- **Enhancing Communication** – Breaking language and distance barriers.
- **Security and Defense** – Safeguarding critical infrastructures.
- **Improving Quality of Life** – Health, education, and beyond.
- **Advancing Science** – Unleashing possibilities beyond imagination.
- **And many more...** – The list keeps growing!

Top Trending Technologies in 2025 and Beyond

- Artificial Intelligence (AI) and Machine Learning (ML)
- Generative AI (GenAI)
- Internet of Things (IoT)
- 5G Technology
- Blockchain and Decentralized Systems
- Quantum Computing
- Cybersecurity and Privacy Solutions
- Augmented Reality (AR) and Virtual Reality (VR)
- Robotic Process Automation (RPA)
- Edge Computing
- Digital Twins
- Biotechnology and Bioinformatics
- Autonomous Systems and Robotics
- Sustainable and Green Technologies
- Metaverse and Web3

Artificial Intelligence (AI)

Compelling Statistic - AI

- Did you know?
- AI is projected to contribute **$15.7** trillion to the global economy by 2030.
- That's more than the combined GDP of China and India!

Real World Examples - AI

- **Doctors (Radiologists)** – Image classification to detect diseases.
- **Chatbots** – Text generation for customer support.
- **Autonomous Vehicles** – Object detection and classification.

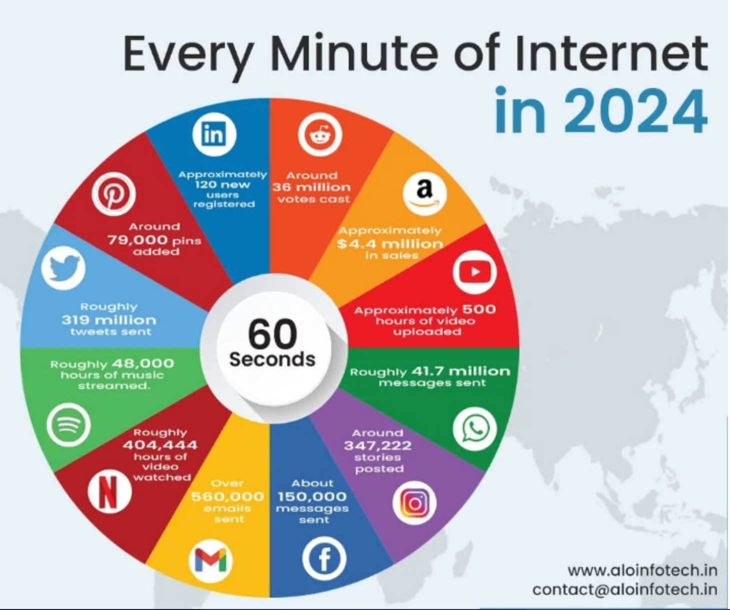
Relatable Scenarios - AI

- Social Media Content Recommendation
- Voice Assistants (Siri, Alexa, Google Assistant)
- Face Unlock on Smartphones
- AI-Powered Navigation (Google Maps, Waze)
- Online Shopping Recommendations
- Spam Email Filtering
- Auto-Correction and Grammar Suggestions

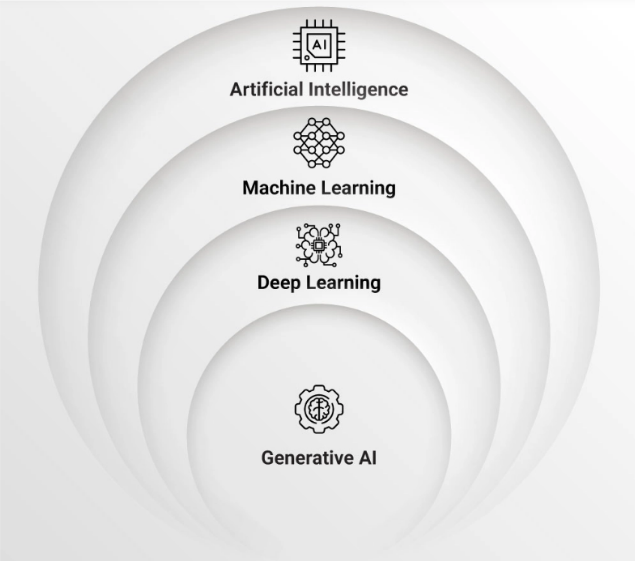
Are We Surrounded by AIR?

No! We are Surrounded by DATA...
...but Starved for INSIGHTS!

Data is the New OIL!



Before Diving into the CORE, we need to understand the BASE!



Artificial Intelligence (AI) - PINNACLE

- AI is the **simulation of human intelligence** by machines to perform cognitive tasks.
- Main Goals of AI:
  - **Mimic Human Intelligence** – Solve complex problems.
  - **Learn and Adapt** – Improve performance over time.
  - **Automate Complex Tasks** – Reduce human intervention.
  - **Perceive and Interact** – Understand images, text, and speech.

Machine Learning (ML)

- ML is a subset of AI where machines **learn from data** to make decisions without explicit programming.
- Domains of ML:
  - Supervised Machine Learning
  - Unsupervised Machine Learning
  - Reinforcement Learning
  - Semi-Supervised Learning

Deep Learning (DL)

- DL uses artificial neural networks (ANNs) to **learn hierarchical features** from large datasets.
- Domains of DL:
  - Computer Vision
  - Natural Language Processing
  - Recommendation Systems
  - Time Series Analysis and Forecasting
  - Transfer Learning

Generative AI (GenAI)

- GenAI creates original content like text, images, audio, and more by learning patterns in data.
- Domains of GenAI:
  - Text Generation and Natural Language Processing (NLP)
  - Image Generation and Computer Vision
  - Audio and Music Generation
  - Video Generation and Synthesis
  - Digital Avatars and Virtual Influencers
  - 3D Model and Digital Twin Generation

Computer Vision (CV)

- CV enables machines **to interpret and analyze visual data** (images/videos) like humans.
- Main Goals of CV:
  - **Image Classification** – Categorize objects.
  - **Object Detection** – Locate and identify multiple objects.
  - **Video Analysis** – Track objects and actions.
  - **Image Segmentation** – Break down images into regions.
  - **Image Restoration/Enhancement** – Improve quality.

Natural Language Processing (NLP)

- NLP helps machines **understand, interpret, and generate human language.**
- Domains of NLP:
  - Text Processing and Tokenization
  - Part-of-Speech (POS) Tagging and Syntactic Analysis
  - Named Entity Recognition (NER)
  - Sentiment Analysis and Opinion Mining
  - Text Classification and Categorization
  - Machine Translation and Multilingual NLP
  - Question Answering and Information Retrieval
  - Speech Recognition and Text-to-Speech (TTS)
  - Text Summarization

Ready to Dive Deeper? Let's Go!

1. Image Classification

What is Image Classification?

Definition

Image classification involves assigning a specific category label to an image based on its visual content.

- Image Classification is a fundamental task in computer vision.
- A machine automatically assigns a label or category to an input image.
- The goal is to analyze and identify patterns in the image and associate it with a predefined class.

Image Classification - Importance

- **Autonomous Vehicles:** Detecting pedestrians, traffic signs, and obstacles.
- **Healthcare:** Diagnosing diseases from medical images.
- **E-commerce:** Classifying products and improving visual search.
- **Security Systems:** Identifying suspicious activities and intruders.

Image Classification - How?

- Goal:
  - Input: An image of a pet.
  - Output: Label ("Cat" or "Dog").
- How It Works:
  - Input Image: Capture an image of a pet.
  - Feature Extraction: Identify patterns like fur texture, ear shape, and eye structure.
  - Model Decision: Based on extracted features, predict whether the image is a cat or a dog.

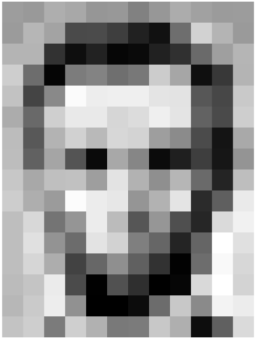
How Does Image Classification Work? - Programmically

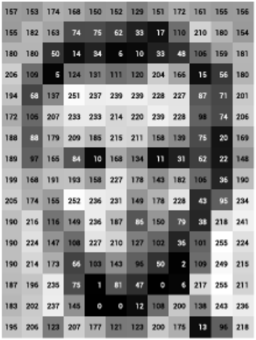
- Pixel Data and Feature Extraction
- Training Models with Labeled Data
- Metrics

1. Pixel Data and Feature Extraction

- Pixel Data:
  - An image is represented as a matrix of pixel values.
  - For grayscale images: Each pixel has an intensity value (0 to 255).
  - For RGB images: Each pixel has 3 channels (Red, Green, Blue).

Image as Pixels



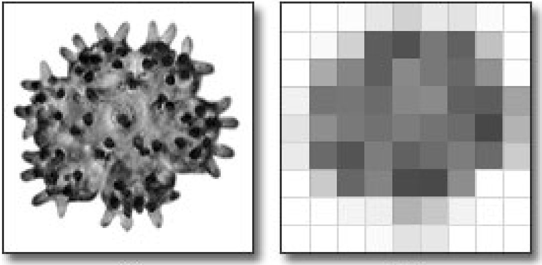


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Gray Scale Image - (0 to 255)

Creation of a Digital Image

Analog Image | **Digital Sampling** | **Pixel Quantization**



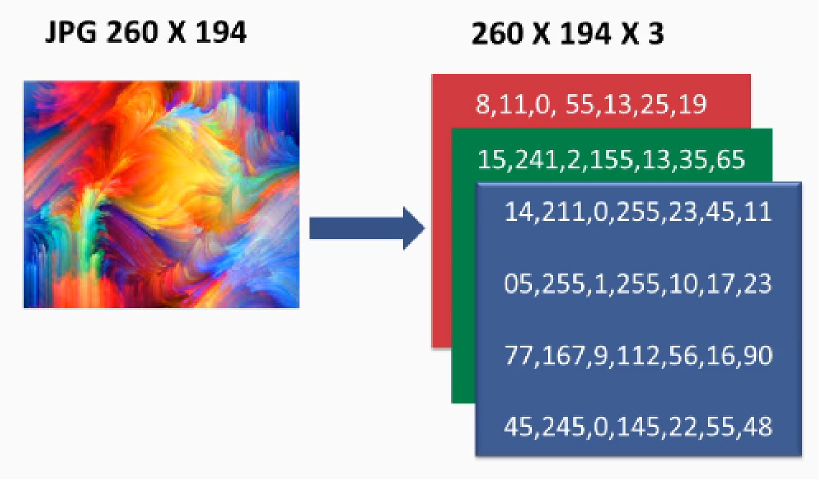
(a) (b)

249	244	240	230	209	233	227	251	255
248	245	210	93	81	120	97	193	254
250	170	133	94	137	120	104	145	253
241	116	118	107	134	138	96	92	163
277	142	121	113	124	115	107	71	179
234	106	84	125	97	108	125	106	204
241	202	102	132	75	73	141	248	252
253	252	244	239	178	199	242	250	245
255	249	244	250	228	231	240	251	253

(c)

Figure 1

RGB Image

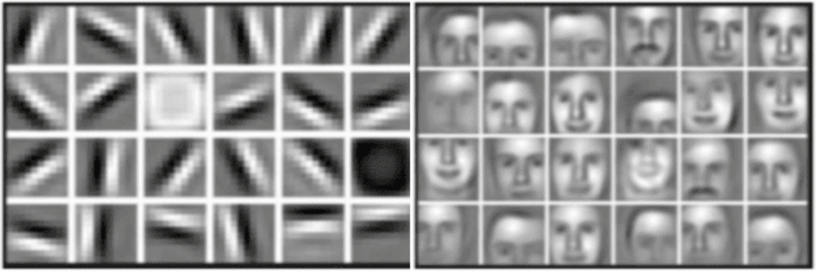


JPG 260 X 194
260 X 194 X 3
8,11,0, 55,13,25,19
15,241,2,155,13,35,65
14,211,0,255,23,45,11
05,255,1,255,10,17,23
77,167,9,112,56,16,90
45,245,0,145,22,55,48

1. Pixel Data and Feature Extraction

- Feature Extraction:
  - Low-Level Features: Edges, corners, and textures.
  - High-Level Features: Object shapes and patterns.

Features



Lines, Corners, Edges   Meaningful Faces

2. Training Models with Labeled Data

- a. Collecting Labeled Data
  - Thousands of images with labels: ("Cat" or "Dog").
  - Labels serve as ground truth for training.
- b. Model Training Process
  - Input: Image and corresponding label.
  - Feature Learning: Model learns patterns to differentiate classes.
  - Loss Calculation: Measures the difference between predicted and actual labels.
  - Backpropagation and Optimization: Adjust model weights to minimize error.

3. Metrics

- Split data into training, validation, and test sets.
- Fine-tune hyperparameters to improve accuracy.

Popular Algorithms for Image Classification

- Convolutional Neural Networks (CNNs)
- Transfer Learning
- Vision Transformers (ViTs)

Deep Dive into CNNs for Image Classification

Why Convolutional Neural Networks (CNNs)?

- CNNs excel at extracting hierarchical patterns from images.
- They reduce computational complexity by leveraging local connectivity and shared weights.
- Ideal for tasks like image classification, object detection, and segmentation.

Neurons (Nodes)

- Neurons are the fundamental building blocks of deep learning models. Each neuron processes input data and produces an output. These neurons are organized into layers.

Layers

- Deep learning models consist of multiple layers of neurons, typically arranged in a sequential fashion.
- The input layer receives data, hidden layers process it, and the output layer produces the final result.
- Common layer types include input, hidden (including convolutional and recurrent layers), and output layers.

Weights and Biases

- Weights and biases are parameters associated with each connection between neurons.
- Weights determine the strength of connections and are adjusted during training to learn patterns in data.
- Biases help neurons capture patterns that may not be apparent from the raw data.

Activation Functions

- Activation functions introduce non-linearity into the neural network, allowing it to model complex relationships.
- Common activation functions include ReLU (Rectified Linear Unit), sigmoid, and tanh.

Loss Function (Cost Function)

- The loss function quantifies how well the model's predictions match the actual target values.
- The goal during training is to minimize the loss function by adjusting weights and biases.

Optimization Algorithm

- Optimization algorithms like stochastic gradient descent (SGD) are used to update the model's weights and biases in a way that minimizes the loss function.
- Variants of SGD, such as Adam and RMSprop, are commonly used.

CNN - Components

- Input layer
- Convolutional layers
- Activation layer
- Pooling (Subsampling) layer
- Fully Connected (Dense) Layers
- Flattening Layer
- Output Layer
- Dropout and Regularization
- Normalization Layers (Batch Normalization)
- Padding
- Strides
- Skip Connections (Residual Connections)

Input Layer

- The input layer receives the raw data, typically in the form of images or grids of data (e.g., pixel values in an image).

Convolutional Layers

- Convolutional layers are the core building blocks of CNNs. They consist of multiple filters (also called kernels) that slide over the input data to extract local features.
- Each filter captures specific patterns or features, such as edges, corners, or textures.
- Convolution operations involve element-wise multiplications and summations between the filter and a region of the input, producing feature maps.

Activation Function (ReLU)

- After each convolution operation, a Rectified Linear Unit (ReLU) activation function is applied element-wise to introduce non-linearity.
- ReLU helps the network learn complex and non-linear patterns in the data.

Pooling (Subsampling) Layers

- Pooling layers are used to downsample feature maps and reduce their spatial dimensions.
- Common pooling methods include max-pooling and average-pooling, which retain the most significant information in the feature maps while reducing computational complexity.

Fully Connected (Dense) Layers

- Fully connected layers are traditional neural network layers in which every neuron is connected to every neuron in the previous and subsequent layers.
- These layers enable high-level feature combinations and are typically used in the later stages of a CNN.

Flattening Layer

- Before connecting the convolutional layers to the fully connected layers, the feature maps are flattened into a one-dimensional vector.

Output Layer

- The output layer produces the final predictions or classifications based on the learned features.
- The activation function in the output layer depends on the task; for example, softmax is commonly used for multi-class classification.

Dropout and Regularization

- Dropout layers may be added to mitigate overfitting by randomly deactivating a fraction of neurons during training.
- Regularization techniques such as L1 or L2 regularization can also be applied to the fully connected layers.

Normalization Layers (Batch Normalization)

- Batch normalization layers help stabilize training by normalizing the inputs to each layer.
- They reduce internal covariate shift and improve the convergence of the network.

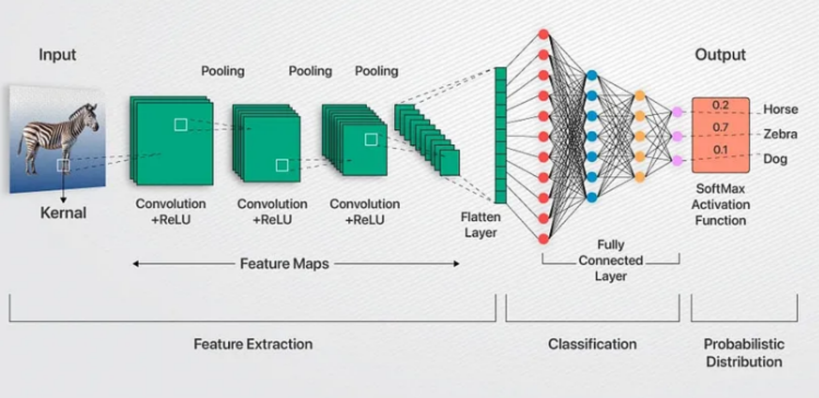
Padding

- Padding is sometimes added to the input data to control the spatial dimensions of feature maps after convolution.
- Zero-padding is a common technique used to maintain spatial information.

Strides

- Strides determine how much the filter moves across the input data during convolution.
- Strides affect the spatial resolution of feature maps

CNN Architecture



Variants of CNN architecture

- LeNet-5
- AlexNet
- VGGNet (VGG)
- GoogLeNet (Inception)
- ResNet (Residual Network)
- MobileNet
- DenseNet (Densely Connected Convolutional Networks)
- EfficientNet
- YOLO (You Only Look Once)
- UNet
- Attention-Based Models (e.g., Vision Transformers) and many more...

Application of CNN architecture

- Image Classification
- Object detection
- Image Segmentation
- Face Recognition
- Gesture Recognition
- Emotion detection
- Medical Imaging
- Video analysis
- Art restoration
- Self-driving cars
- Document analysis and many more...

2. Text Generation

What is Text Generation?

- Text generation involves producing coherent, meaningful text based on a given input or context.
- It is widely used in chatbots, content generation, and automated report writing.
- Modern models generate human-like text by predicting the next word or sequence.

Understanding Natural Language Generation (NLG)

- **Definition:** NLG is a subfield of NLP that transforms structured data into natural language.
- **Goal:** Generate grammatically correct and contextually meaningful text.
- **Applications:**
  - Automated news reports.
  - Personalized marketing content.
  - Summarizing structured data in real time.

Difference Between Text Classification and Generation

- **Text Classification:** Assigns a label or category to a given text.
- **Text Generation:** Produces text based on a prompt or initial input.
- **Key Difference:** Classification predicts labels, while generation predicts and generates sequences.

Sequence Models and Their Evolution

- **Recurrent Neural Networks (RNNs)**
  - Processes sequential data by maintaining hidden states.
  - Challenges: Vanishing gradient problem and limited long-term memory.
- **Long Short-Term Memory (LSTM)**
  - Enhanced version of RNNs with memory cells and gates.
  - Capable of capturing long-term dependencies in text.
- **Gated Recurrent Units (GRUs)**
  - Similar to LSTM but with fewer gates, making it computationally efficient.
  - Suitable for tasks requiring faster training and lower complexity.

Exploring Transformers and Attention Mechanisms

The Rise of Transformers

- Transformers revolutionized NLP by introducing parallel processing and better handling of long-range dependencies.
- Introduced in the paper **“Attention Is All You Need”** (2017).
- Widely adopted in state-of-the-art models for text, image, and multimodal tasks.

What is the Transformer Architecture?

- **Encoder-Decoder Structure:**
  - **Encoder:** Processes the input sequence and generates a contextual representation.
  - **Decoder:** Generates the output sequence based on the encoder's context.
- **Parallelization:** Enables training on large datasets by processing tokens simultaneously.
- **Key Innovation:** Utilizes self-attention mechanisms for capturing global dependencies.

Importance of Self-Attention Mechanism

- **Self-Attention:** Allows the model to weigh the importance of different words in a sequence.
- **How It Works:**
  - Each word attends to all other words in the sequence.
  - Generates attention scores to focus on relevant parts of the input.
- **Benefits:**
  - Captures long-term dependencies efficiently.
  - Reduces computation time compared to traditional RNNs.

Popular Models Based on Transformer Architecture

- **BERT (Bidirectional Encoder Representations from Transformers)**
  - Pre-trained using Masked Language Modeling (MLM).
  - Provides contextual embeddings for downstream NLP tasks.
  - Applications: Sentiment analysis, question answering, and text classification.
- **GPT (Generative Pre-trained Transformer)**
  - Trained in an autoregressive manner to predict the next token.
  - Generates coherent and context-aware text.
  - Applications: Chatbots, content generation, and creative writing.
- **T5 (Text-to-Text Transfer Transformer)**
  - Converts all NLP tasks into a text-to-text format.
  - Fine-tuned for multiple NLP tasks using a unified approach.
  - Applications: Text summarization, translation, and question answering.

Text Generation with GPT Models

How GPT Models Work

- **Pre-training:**
  - Model learns to predict the next word in a sentence using a massive corpus of text.
  - Trained in an autoregressive manner to generate coherent and context-aware text.
- **Fine-tuning:**
  - Pre-trained model is adapted for specific tasks using labeled data.
  - Fine-tuning enhances performance for downstream tasks like summarization and Q&A.

Tokenization and Decoding Methods

- **Tokenization:**
  - Splits text into smaller units called tokens (subwords, words, or characters).
  - Common methods: Byte Pair Encoding (BPE), WordPiece, and SentencePiece.
- **Decoding Methods:**
  - **Greedy Search:** Selects the most probable token at each step.
  - **Beam Search:** Explores multiple possible sequences to find the best result.
  - **Top-k Sampling:** Samples from the top-k most likely tokens.
  - **Top-p (Nucleus) Sampling:** Selects tokens with cumulative probability above a threshold.

Real-World Applications of GPT Models

- **Chatbots (ChatGPT)**
  - Engages in natural, human-like conversations.
  - Widely used in customer support, virtual assistants, and FAQ bots.
- **AI-Powered Content Generation**
  - Generates articles, blog posts, product descriptions, and marketing content.
  - Assists writers by providing coherent text suggestions.
- **Code Generation (Codex)**
  - Generates code snippets from natural language prompts.
  - Accelerates software development with AI-assisted coding.
  - Powers platforms like GitHub Copilot.

Challenges and Limitations

Challenges and Limitations

Challenges in Image Classification

- **Overfitting and Model Bias**
  - Model memorizes training data, leading to poor generalization.
  - Biases in training data can affect model predictions.
- **Handling Noisy and Unstructured Data**
  - Presence of irrelevant information affects model accuracy.
  - Requires data preprocessing and augmentation for noise reduction.

Challenges in Text Generation

- **Coherence and Context Retention**
  - Models may generate grammatically correct text but lose coherence over long passages.
  - Difficulty in maintaining logical consistency across paragraphs.
- **Ethical Concerns and Bias in AI**
  - Biases in training data can result in unfair or offensive content.
  - Ethical concerns regarding misinformation and plagiarism.

mail me: er.anandprem@gmail.com / premanand.s@vit.ac.in
ring me: +91 73586 79961
follow me: Linkedin
author at Analytics Vidhya: premanand17
author at Medium: Premanand S

Predicting the future isn't magic, it's artificial intelligence!