1. Problem Definition

1. Problem Statement

2. Ideal Problem Solution

3. Understanding and insight into the problem

4. Technical requirements

Associated roles: IT Business Analyst

# 2. Research

1. Data Structure and Source

2. Solution form

3. Neural Network / Model Architecture

4. Algorithm Research

5. Hardware Requirements

6. Software Requirements

Associated roles: Machine Learning Researcher, Data Scientist, AI Researcher

# 3. Data Aggregation / Mining / Scraping

Data understanding is paramount and any sourced data should be examined and analyzed utilizing visualization tools or statistical methods

* The data gathered needs to be diverse enough to ensure that the model predictions capabilities accommodate a variety of possible scenarios.
* The data gathered needs to aspire to be unbiased to ensure that the model can generalize appropriately during inference.
* The data gathered needs to be abundant.

Tools for collecting data will vary. Data sources could come in the form of APIs, XML feeds, CSV, or Excel files. In some scenarios, data mining/scraping from online sources is required. Ensure to check on third party websites scraping/mining policies before conducting scrape

Associated roles: Data Scientist, Data Analyst

# 4. Data Preparation / Preprocessing / Augmentation

Data preprocessing could include the identified steps below, but not limited to the mentioned steps:

* Data Reformatting (resizing images, modification to color channels, noise reduction, image enhancement)
* Data Cleaning
* Data Normalisation

Data augmentation is a step that is carried out to improve the diversification of data that has been sourced. Augmentation of image data could take the following forms:

* Rotation of an image by any arbitrary degrees
* Scaling of an image either to create zoomed in/out effects
* Cropping of an image
* Flipping (horizontal or vertical) of an image
* Mean Subtraction

Associated roles: Data Scientist

# 5. Model Implementation

You rarely have to implement a model from scratch. The following might be expected to be conducted during the model implementation stage:

* Removal of last layers within a neural network to repurpose models for specific tasks. For example, removing the last layer of a Resnet neural network architecture enables the utilization of a descriptor provided by the model within an encoder-decoder neural network architecture
* Fine-tuning pre-trained models

Associated roles: Data Scientist, Machine Learning Engineer, Computer Vision Engineer, NLP Engineer, AI Engineer

# 6. Training

# The training of the implemented model involves iteratively passing mini-batches of the training data through the model for a specified amount of epochs

**Hyperparameters**: These are values that are defined before the training of the network begins; they are initialized to help steer the network to a positive training outcome. Their effect is on the machine / deep learning algorithm, but they are not affected by the algorithm. Their values do not change during training. Examples of hyperparameters are regularization values, learning rates, number of layers, etc.

**Network parameter**: These are components of our network that are not manually initialized. They are embedded network values that are manipulated by the network directly. An example of a network parameter is the weights internal to the network.

When conducting training, it is vital to ensure that metrics are recorded of each training process and at each epoch. The metrics that are generally collected are the following:

* Training accuracy
* Validation accuracy
* Training Loss
* Validation Loss

To collate and visualize training metrics, tools such as visualization tools Matplotlib and Tensorboard can be utilized.

By visualizing the training metrics, it is possible to identify some common ML model training pitfalls, such as underfitting and overfitting.

* **Underfitting**: This occurs when a machine learning algorithm fails to learn the patterns in a dataset. Underfitting can be fixed by using a better algorithm or model that is more suited for the task. Underfitting can also be adjusted fixed by recognizing more features within the data and presenting it to the algorithm.
* **Overfitting**: Overfitting can occur if the training data does not accurately represent the distribution of test data. Overfitting can be fixed by reducing the number of features in the training data and reducing the complexity of the network through various techniques

# 7. Evaluation

Some examples of evaluation strategies that can be leveraged are as follows:

**Confusion matrix** (error matrix): Provides a visual illustration of the number of matches or mismatches the annotation of the ground truth to the classifier results. A confusion matrix is typically structured in tabular form, where the rows are filled with the observational results from the ground-truth, and the columns are filled with inference results from the classifier.

**Precision-Recall:** These are performance metrics that are used to evaluate classification algorithms, precision captures the number of results returned that are relevant, while recall captures the number of relevant results in your dataset that are returned.

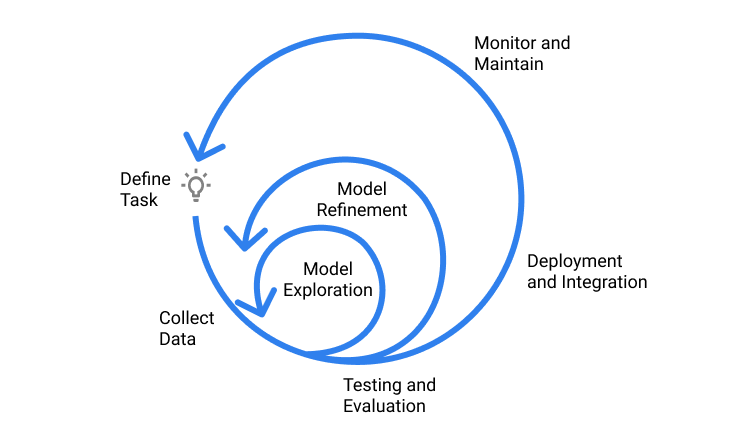
# 8. Parameter tuning and Inference

# 9. Model Conversion to appropriate mobile format

# 10. Model Deployment

Model refinement techniques to avoid underfitting and overfitting like:

1. Controlling hyperparameters
2. Regularisation
3. Pruning



Define the task : - evaluation criteria, optimization function, and loss function, data collection / data generation process,