VISUAL INSPECTION IN THE MANUFACTURING INDUSTRY USING AI

August 25, 2023

- BY VANSH TIWARI,

GANDRAKOTA SAI LALITHA SHALINI,

DEEPAK YADAV,

DHARSHINI SANKAR RAJ

1. Abstract

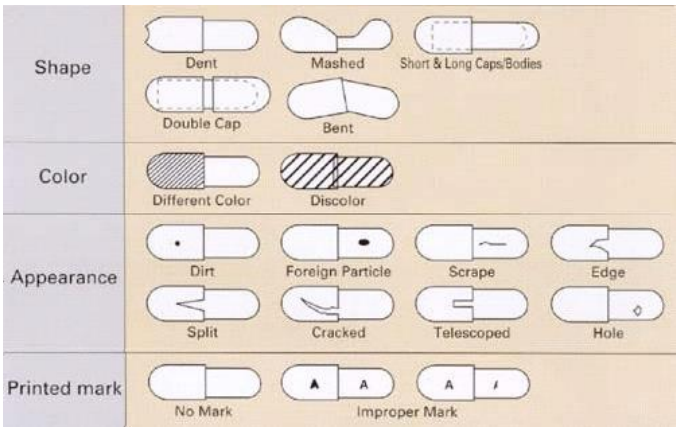
- Visual inspection plays a crucial role in ensuring the quality and reliability of products in manufacturing industry like Bisleri, Coca-Cola, Apollo or small pharmaceutical manufacturing plants. Traditional inspection techniques use man labour which is time consuming, labour-intensive and prone to human error. The help of an autonomous system consists of AI will help in inspection by saving time and cost of the manufacturing company.
- The product is a software type of a product which can be operated through any good camera, it can scan the units and find defects using DL/ML model.

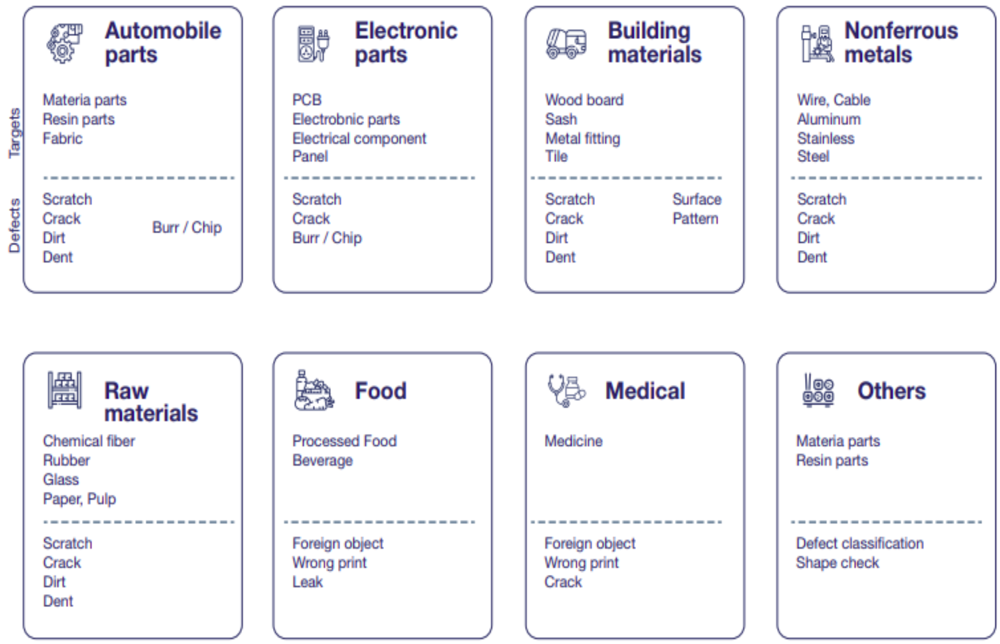
2. Problem statement

- For companies that manufacture cars, semiconductor chips, medicines, etc. for them, production quality and efficiency are two of the best performance measures in the industry. The poor manufacturing quality of the product can lead to high operational and financial costs, which are caused by defects in the machinery in the form of reworked parts, scrap, reduced yield, increased work in processing inventory, post-sale recalls, warranty claims and repairs.

- Many production company releases defect units due to lack of proper surveillance like broken can, damaged medicine capsule, broken plastic toys, irregular shape or colour of bottles and cans etc.

Below are images of types of defects in figure 2.1, 2.2 respectively





3. Market/Customer/Business need Assessment

- In the autonomous era every country wants to earn by using machines which can work 24/7 to get brilliant production so they can export it to other countries. In covid-19 period there was a huge demand of masks, sanitizers, vaccines, syringe, medicines so there was a load on production companies. One of the crucial objectives is not to deliver defective product so instead of using man power we can use a computer vision supervision system which can detect the defects and alert the machine to stop production.

1. The system only needs a good quality camera, a monitor to show the defective piece and an AI model.
2. Faster than human supervision.
3. Can work 24/7.
4. Defect less product delivery which is good for companies' reputation.
5. Light system in size and weight.
6. Easily usable by workers at manufacturing plant.

4. Target Specification

- This product has application of object detection and measuring the dimensions and colour of object like height, breadth, shape, circumference etc. The target will be small manufacturing companies or companies which have labour intensive manufacturing plants. Each companies have a specific product like bottles, capsule of medicine, cans etc, after training the ML/DL model on that specific product we can supervise the upcoming products of that company.

5. External Search (Information/statistics)

- Visual inspection AI is a technology that uses AI and computer vision to automate inspections in industries like manufacturing.
- The process involves capturing images or videos of products or materials using cameras or sensors.

- Visual inspection AI in manufacturing has several benefits including:
  - **Increased accuracy:** You can achieve high accuracy and consistency in defect detection by utilizing computer vision applications for visual inspection. It is often challenging to accomplish through human inspection alone. This results in fewer false positives and negatives that allows manufacturers to identify and address issues more efficiently.

    *For example, a **semiconductor manufacturer in Taiwan** reported a 10% reduction in scrap rates and a 50% increase in throughput after implementing a visual inspection AI system.*
  - **Reduced costs:** Automating the inspection process with visual inspection AI can help you lower labour costs related to manual inspection. Additionally, catching defects early in production can result in lower scrap rates and rework, saving on materials and resources.

*For example, a **manufacturer of automobile parts in Japan** reduced labour costs by 30% and achieved a 95% defect detection rate by using a visual inspection AI system.*

- **Improved efficiency:** Implementing computer vision systems for quality control in manufacturing can have significant benefits. AI can boost manufacturing efficiency with predictive maintenance, quick inspections by quality QC and reduce downtime.

  Using AI-powered computer vision, you can easily identify and address defective products, meet production demands and maintain high-quality standards. This ensures your manufacturing processes run smoothly with minimal interruptions.

  *For example, a **packaging manufacturer in the US** reported a 50% reduction in inspection time and a 10% reduction in labour costs after implementing a visual inspection AI system.*
- **Enhanced safety:** In some cases, visual inspection AI can reduce the need for manual inspection by quality managers in hazardous environments, improving workers safety.

  *For example, the **BMW Din golfing plant in Germany** uses Visual AI to monitor the factory floor for safety hazards. The system detects when workers aren't wearing required PPE or machines operate outside normal parameters and alerts the safety team. This has helped reduce accidents, injuries and increase efficiency.*

Some of the real-world examples are as follows:

- **Foxconn:** The world's largest electronics manufacturer implemented visual inspection AI and reduced inspection time by 30%, while improving accuracy by 80%.
- **Bosch:** The German engineering and technology company implemented computer vision and machine learning algorithms to inspect automotive parts and achieved a 10% increase in production efficiency.
- **GE:** The American multinational conglomerate uses visual inspection AI to detect defects in aircraft engines and other critical components. This has led to a 25% reduction in inspection time and a 30% reduction in manufacturing costs.

- **Flex:** The Singaporean electronics manufacturer uses computer vision and machine learning algorithms to inspect printed circuit boards. This has resulted in a 90% reduction in inspection time and a 99% reduction in false positives.
- **Siemens:** The German multinational conglomerate uses AI to inspect wind turbine blades for defects. This has led to a 30% increase in inspection accuracy and a 50% reduction in inspection time.

Data set for iron casting defect: https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product

6. Benchmarking and patents

- **Comparison with non-Ai technology**:
  Multiple visual inspections are typically conducted at various points in the typical manufacturing process. Visual inspection is usually a time consuming and error prone process because it is highly manual, requiring intent focus from human laborers over extended periods of time something most people aren't particularly good at. Some common problems facing traditional approaches to visual inspection include:

    1. Product changes necessitate frequent reconfiguration of traditional inspection machinery, which is inflexible and difficult to adapt in fast-paced environments.
    2. Manual inspectors vary based on their experience and the limitations of human perception, which can cause inconsistent quality control.
    3. Error rates for human inspectors are estimated at 20-30%. Conversely, McKinsey research states that AI-based visual inspection can lead to a 50% increase in productivity and a 90% improvement in defect detection accuracy.

- Comparison between existing AI technology:
    1. Comparison with other AI vision supervision system can include difference in accuracy that is by using different data, DL model, Pipeline etc.

2. Potentiality of business model which can include onetime sale and service or can have key and subscription system i.e., system can work and can have services only after monthly/yearly subscription.
3. Using sensor or different quality cameras which can increase accuracy of prediction.

- Applicable patents are provided in below link:

  https://patents.google.com/patent/CN108463874B/en?q=(vision+inspection+manufacturing+plants+ai)&oq=vision+inspection+in+manufacturing+plants+using+ai

7. Applicable constrains

- **Technical issues:** Implementing visual inspection AI requires expertise in machine learning and computer vision, which may be outside the scope of some manufacturing companies.
- **Quality and accuracy of data:** One of the major challenges in implementing visual AI is ensuring the manufacturing quality and accuracy of the visual data being used for training the algorithms.
- **Integration with existing systems:** Integrating visual AI into manufacturing systems can be challenging due to complexity and time consumption.
- **Cost:** Visual AI implementation can be expensive, especially for smaller manufacturers.
- **Ethical concerns:** There are ethical concerns associated with the use of visual AI, such as potential bias and privacy issues.
- **Deployment and service:** Deploying the system and updating the training data after a particular time period and placing hardware such as camera and monitor is important.

8. Applicable regulations

- Patent for ML/DL algorithm.

- Permission from manufacturing company to use their necessary. data and to understand their system of production.
- Government laws for the product.
- Service and taking care of camera or sensors which will give the input data.
- Modification of system according to company need.
- System design patent.
- Making sure to not damage or change the manufacturing plant environment while integrating.

9. Business opportunity

- This product is mainly used by big companies so we can sell this to small business companies at low cost. The company can also reduce their cost by reducing labours and instead just one supervisor is enough for monitoring the system, we can apply multiple cameras at multiple conveyor belts and all can by monitored by one screen and one supervisor. Also, by selling variation of models like camera, camera + sensor, variation in algorithms etc we can sell product according to their need and budget.
- Vision inspection using AI is preferred over manual inspection because AI gives more accuracy, non-stop working, updating according to new product, greater coverage of refractive index than human eyes, can detect thermal, radioactive wave and other waves using sensors, cameras combination.

10. Concept Generation

- This product will be in form of a software and comes under monitoring/supervising system, one of the main aims of this product is to save resources and so give client a defect free production.

- The defective product production rate is almost 15% and this item can be stopped to sell by using accurate vision inspection method like using computer vision, sensor system we can get more advantage than traditional methods.

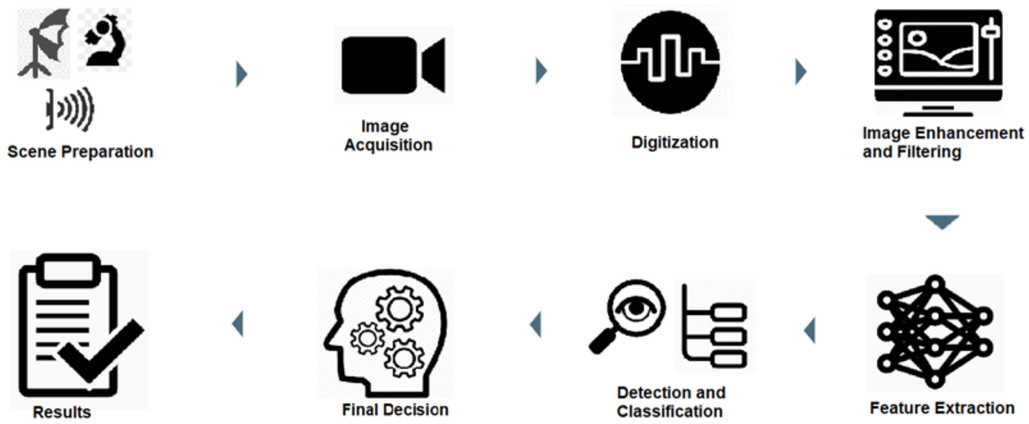
11. Concept development and product overview

- **List of software requirements:**
  Python 3(IDE: Jupyter Notebook).

- **List of hardware requirements:**
  Intel Core i5 10th gen/AMD Ryzen 7 5800H.
  8GB RAM.
  Windows 11.
  Good quality camera.
- **Product details and working:**
  This visual inspection systems can be trained to recognize patterns and anomalies in images and videos and can be used to identify defects that are difficult for humans to see. It can also automate the inspection process, helping companies reduce the time and cost of manual inspection.

  In manufacturing, visual inspection examines products or materials for defects or deviations from specifications using the human eye or a machine vision system. Artificial intelligence (AI) can automate and improve visual inspection accuracy by analysing images or video streams to identify defects or deviations.

  Below is basic diagram of system workflow that is from where it will take input and analyse the target to give output



System overview Fig 11.1

image source:-(https://images.app.goo.gl/dwPGVct6BHMNkwWL6)

A) Feasibility:

The feasibility of this system depends on several factors, such as the availability of data, the complexity of the product, the cost of implementation, and the expected benefits. The system has a high potential to improve the quality and efficiency of manufacturing processes by automating visual inspection using AI. However, there are also some challenges and limitations that need to be addressed, such as technical issues, ethical concerns, integration with existing systems, and applicable regulations. Therefore, it is suggested to conduct a thorough market research and feasibility analysis before developing and deploying this system.

B) Viability:

There's a clear market demand for efficient quality control in manufacturing. The system's ability to save costs and time, along with customization options, positions it well. Real-world successes of similar systems and the technology advantage enhance viability. However, challenges like initial investment and technical expertise should be addressed. With careful planning and validation, the system has a promising chance to succeed in the market.

C) Monetization:

Monetization is the process of generating revenue from something that does not normally produce income. For this visual inspection AI system, which is a software solution that automates and improves the quality control process for manufacturing customers, monetization could be achieved by various methods, such as:

- Charging a fee for using the system or accessing certain features or tools.
- Displaying ads or sponsored content within the system interface or the inspection results.
- Offering premium subscriptions or memberships that provide additional benefits or services.
- Selling data or insights derived from the user interactions or feedback.
- Partnering with other platforms or businesses that can benefit from the system or the inspection data.

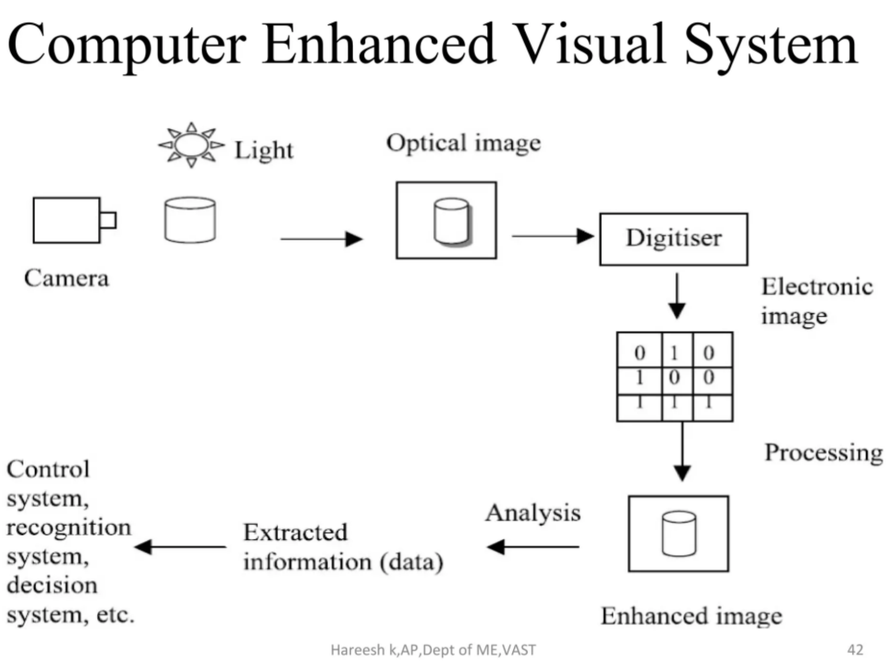
- **Methodology:**

    - Using ML and DL algorithms (SVM, CNN, AlexNet, GoogleNet etc.) will be the main focus of tackling the problem.
    - Problems regarding the measurement of products can be solved using Visualization.
    - The main factors of these defects are length, breadth, area, perimeter, quality, the colour of the product, quality of the raw materials, etc.
    - The benefit of ML and DL automation is that providing real-time data on particular industrial machinery will improve the machinery's accuracy of the machinery resulting in a self-resolving and enhancing product quality supplied by a machine.
    - This method will also provide us with a dataset that can be used on other similar machines or for new advancement of current versions of the machinery.

- **Workflow:**

  Depending on the industry where it being use, the physical equipment can actually be categorized into three subsystems.

  - **Feeding system**—Spreads items evenly and moves them at a constant speed, so that the optical system could capture frames of individual items.
  - **Optical system**—Consists of a specifically adjusted lighting source and a sensor (usually, a digital camera). The optical system captures images of inspected items so that the software can process and analyse them.
  - **Separation system**—Removes defective items and/or grades and separates products into several categories according to their quality.



Detailed overview of how software model works Fig 11.2

image source (https://www.slideshare.net/hareeshkodanghat/introduction-to-ndt-and-visual-inspection)

12. Code Implementation

- For implementing code, I have taken iron casting defect examples and made a simple custom CNN for supervision predict
- Below are the necessary codes

```
In [1]: import tensorflow as tf
        import keras
        import seaborn as sns
        from kerastuner.tuners import RandomSearch
        from kerastuner.engine.hyperparameters import HyperParameter as hp
        from keras.layers import Dense,Dropout,Activation,Add,MaxPooling2D,Conv2D,Flatten,BatchNormalization,MaxPool2D
        from keras.models import Sequential
        from keras import layers
        from keras.preprocessing.image import ImageDataGenerator
        import numpy as np
        import matplotlib.pyplot as plt
        from keras.preprocessing import image
        import random
        from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping

        C:\Users\Vansh\AppData\Local\Temp\ipykernel_11612\2589193303.py:4: DeprecationWarning: `import kerastuner` is deprecated, pleas
        e use `import keras_tuner`.
          from kerastuner.tuners import RandomSearch
```

```
In [32]: batch_size = 64
         epochs = 10
         img_height = 299
         img_width = 299
         img_size = (img_height, img_width)
         dataset_url = r"C:\Users\Vansh\Downloads\casting\casting_512x512\casting_512x512"
         data_dir = dataset_url
         data_dir

Out[32]: 'C:\\Users\\Vansh\\Downloads\\casting\\casting_512x512\\casting_512x512'
```

- importing library and data

```
In [33]: seed = 0
         random.seed(seed)
         np.random.seed(seed)
         tf.random.set_seed(seed)
         train_set = tf.keras.utils.image_dataset_from_directory(
             data_dir,
             validation_split=0.2,
             class_names =  ['ok_front', 'def_front'],
             subset="training",
             seed=seed,
             image_size=(img_height, img_width),
             batch_size=batch_size)
         val_set = tf.keras.utils.image_dataset_from_directory(
             data_dir,
             validation_split=0.2,
             class_names =  ['ok_front', 'def_front'],
             subset="validation",
             seed=seed,
             image_size=(img_height, img_width),
             batch_size=batch_size)

         Found 1300 files belonging to 2 classes.
         Using 1040 files for training.
         Found 1300 files belonging to 2 classes.
         Using 260 files for validation.
```

- For the sake of implementation, I have used only 1300 files not all and used validation and train set split

```
In [38]: c_model = Sequential([
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(1, activation = 'sigmoid')
])
```

```
In [39]: c_model.compile(optimizer='adam',
                     loss='binary_crossentropy',
                     metrics=['accuracy'])
```

```
In [40]: class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if logs.get('accuracy') == 1.0 and logs.get('val_accuracy') == 1.0 :
            print("\nReached 100% accuracy so cancelling training!")
            self.model.stop_training = True

terminate_callback =  myCallback()
```

```
In [41]: reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.5, verbose=1,min_delta=0.01,
                                                patience=5, min_lr=0.001)
```

- This is the simple CNN model which using callbacks and learning rate control over epochs

```
: history = c_model.fit(train_set,
                        validation_data=val_set,
                        epochs=epochs,
                        callbacks= [reduce_lr, terminate_callback]
)
```

```
Epoch 1/10
17/17 [==============================] - 78s 4s/step - loss: 2.8968 - accuracy: 0.5644 - val_loss: 0.9176 - val_accuracy: 0.588
5 - lr: 0.0010
Epoch 2/10
17/17 [==============================] - 73s 4s/step - loss: 0.6159 - accuracy: 0.6846 - val_loss: 0.5379 - val_accuracy: 0.780
8 - lr: 0.0010
Epoch 3/10
17/17 [==============================] - 74s 4s/step - loss: 0.5176 - accuracy: 0.7490 - val_loss: 0.4616 - val_accuracy: 0.807
7 - lr: 0.0010
Epoch 4/10
17/17 [==============================] - 73s 4s/step - loss: 0.4490 - accuracy: 0.7827 - val_loss: 0.5125 - val_accuracy: 0.765
4 - lr: 0.0010
Epoch 5/10
17/17 [==============================] - 75s 4s/step - loss: 0.4397 - accuracy: 0.7875 - val_loss: 0.4310 - val_accuracy: 0.838
5 - lr: 0.0010
Epoch 6/10
17/17 [==============================] - 74s 4s/step - loss: 0.3987 - accuracy: 0.8010 - val_loss: 0.5438 - val_accuracy: 0.788
5 - lr: 0.0010
Epoch 7/10
17/17 [==============================] - 73s 4s/step - loss: 0.3870 - accuracy: 0.8221 - val_loss: 0.3616 - val_accuracy: 0.803
8 - lr: 0.0010
Epoch 8/10
17/17 [==============================] - 74s 4s/step - loss: 0.3148 - accuracy: 0.8567 - val_loss: 0.3068 - val_accuracy: 0.846
2 - lr: 0.0010
Epoch 9/10
17/17 [==============================] - 79s 5s/step - loss: 0.2701 - accuracy: 0.8885 - val_loss: 0.2745 - val_accuracy: 0.884
6 - lr: 0.0010
Epoch 10/10
17/17 [==============================] - 75s 4s/step - loss: 0.2090 - accuracy: 0.9154 - val_loss: 0.2221 - val_accuracy: 0.907
7 - lr: 0.0010
```

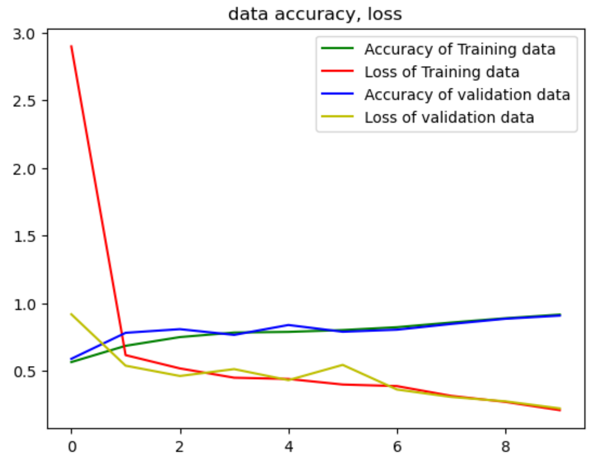
- I have only implemented 10 epochs for the sake of implementation as my laptop didn't have resources and was taking time

```
In [49]: get_ac = history.history['accuracy']
         get_los = history.history['loss']
         val_acc = history.history['val_accuracy']
         val_loss =history.history['val_loss']
```

```
In [50]: epochs = range(len(get_ac))
         plt.plot(epochs, get_ac, 'g', label='Accuracy of Training data')
         plt.plot(epochs, get_los, 'r', label='Loss of Training data')
         plt.plot(epochs, val_acc, 'b', label='Accuracy of validation data')
         plt.plot(epochs, val_loss, 'y', label='Loss of validation data')

         plt.title('data accuracy, loss')
         plt.legend(loc=0)
         plt.figure()

Out[50]: <Figure size 640x480 with 0 Axes>
```




```
<Figure size 640x480 with 0 Axes>
```

- the evaluation is done by using accuracy, loss in training and validation set

13. Conclusion

- Ai technology is still in developing phase and a wise business strategy is to invest in developing technology because you never know where the limit of a technology will be. This product is a mixture of hardware (camera, monitor, sensor) and software and can be deployed to cloud and delivered in for of software to the small business manufacturing companies
- The team required will be:
  - Supervisor
  - Data engineer
  - ML/DL engineer
  - Cloud Engineer (if data is too large than it would be better to deploy on cloud)
  - Software engineer

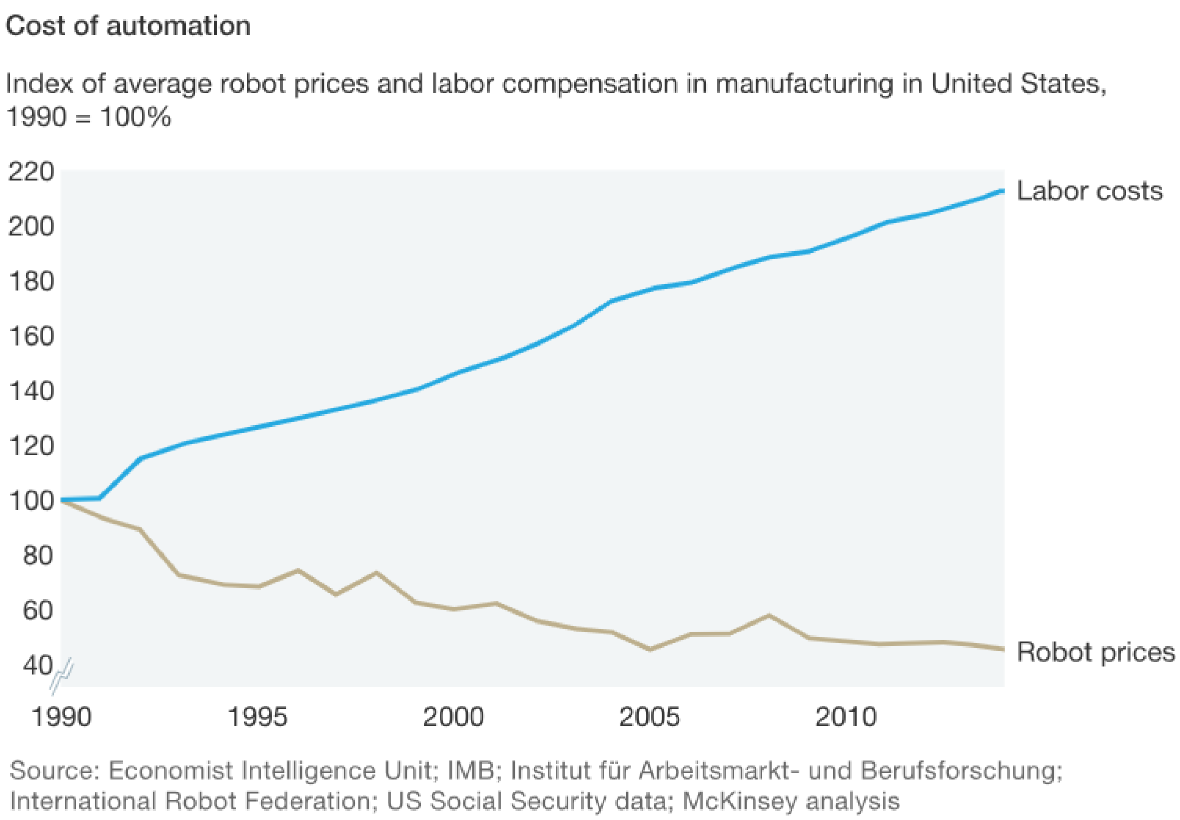
- At the conclusion this product can save time, resource of company and can be sold in varieties of models which client can buy according to their need like in production of gas we can use thermal camera to warn them or to check if the container is leaking, on the normal temperature or not

14. References

- https://www.softwebsolutions.com/resources/visual-inspection-ai-in-manufacturing.html#:~:text=Visual%20inspection%20AI%20is%20a,materials%20using%20cameras%20and%20sensors.
- https://nanonets.com/blog/ai-visual-inspection/
- https://cloud.google.com/blog/products/ai-machine-learning/improve-manufacturing-quality-control-with-visual-inspection-ai

15. Business Model

- For this product subscription-based business model is the most efficient to implement and will be benefit for both client and our side.
- We can offer different subscription-based packages to client, in that way they can also trust our service and understand the benefits of our product
- We can charge them either hourly charge or monthly charge.
- The monthly packages can be of 1.10 lakh for 2 months, 2.15 lakh for 4 months, 11.5 lakh for 12months.
- The hourly charge can be range from Rs. 50 to Rs. 250 based on the requirement of their product inspection.
- If the inspection target is small (microchip, syringe, capsule, etc) then we will need high quality camera or macro camera to get better results. It can increase the cost of the client. While if the target is big enough (Toy, Bottles, tube lights, etc) that can be detected easily by camera then it will decrease the cost of client.



- If the manufacturer manufactures multiple types of products in one production belt, then the model should be trained to identify defects in multiple types of samples in one go.
- For this problem we can train model on cloud and store data on cloud which will reduce execution time and will not take much amount of storage. This particular case will need a Cloud engineer or a specialized ML engineer which can handle cloud also, in this case we will need to extend cost as engineer will do more work or an additional cloud engineer will be there.

16. Business Equation

- The Equation of business is done on linear line equation.
- The equation's **profit** will depend on selected package that is done by the company and the **cost** will depend upon the number of resources used for the project.
- The cost is denoted by ***C***
    - The value of C will be calculated by number of **engineers** for the project and the type of **camera** used of the target product

- For Camera it can range from Rs. 5,000 to 15,000, it will be denoted by ***ca***
- The link of camera: https://www.imperx.com/vision-inspection-cameras/
- https://alphatechsys.co.in/basler/?gclid=CjwKCAjwloynBhBbEiwAGY25dLMu7jphyeAVkDVOKiZoQnMdIIJCpn9FMaBEFyCbpuLyoy2doTfjPRoCZy0QAvD_BwE
- The minimum type engineers required will be 2 Machine Learning engineer denoted by ***ml*** and a software engineer denoted by ***se.***
- Additionally, if data is too large then to reduce cost of computing and to optimize data storage, we would require a cloud engineer denoted by ***ce.***

- The profit will be denoted by ***Y*** the package or sale will be denoted by ***P*** which will be mean of packages**.**
- So, the equation will be:

  $Y = P*x(t) - C$

  Where $C = (2*ml + ca + se + K*ce)$ *here K will be 0 or 1 depending upon the project.*

- ***x(t)*** is the time function which shows profit at time ***t***.
- The graph will show positive slope as after deploying the product there won't be any frequent maintenance.

Financial modelling:

Visual Inspection in the manufacturing industry using AI involves multiple components, including revenue, costs, and potential benefits and considering how market factors can impact both revenue and costs.

Simplified equation –

$$Total\ Benefits=\sum_{i=1}^{n}\left(Market\ Trend_i\times\left(\left(a\times Revenue_i\right)-\left(b\times Cost_i\right)\right)\right)$$

- n is the number of inspection instances.
- Market Trend represents the market trend factor for instance i.
- a and b are coefficients to adjust the relationship between revenue and costs.

Market Trend Factor: The market trend factor should capture how overall market conditions influence the benefits of AI-driven visual inspection. This could be a qualitative assessment based on factors such as industry growth, technological advancements, competitive landscape, and customer demand depends on industry to industry and work. Assign a value between 0 and 1, where 1 indicates the most favorable market conditions and 0 indicates highly unfavorable conditions.

Revenue Potential: The revenue potential can come from various sources:

- Reduced Defects and Rework: Calculate the cost savings from identifying defects early, reducing rework, and improving product quality.
- Increased Throughput: Estimate revenue growth due to increased production throughput resulting from faster inspection.
- Customer Satisfaction: Consider the potential for improved customer satisfaction leading to higher retention rates and referrals.
- Premium Pricing: If AI-driven inspection enhances product quality, it might enable premium pricing.

$$\text{Base Revenue}_i=(\text{Reduced Defects}_i+\text{Increased Throughput}_i+\text{Customer Satisfaction}_i+\text{Premium Pricing}_i)$$

$$\text{Revenue}_i=\text{Base Revenue}_i\times(1+\text{Growth Rate}_i)^{\text{Years}_i}$$

- Base Revenue - is the estimated initial revenue from AI-driven inspection for instance i.
- Growth Ratei represents the expected growth rate of revenue for instance i.

Cost of Implementation:

- Technology Development: Costs related to developing or acquiring the AI system, including software, hardware, and licensing fees.
- Data Collection and Processing: Expenses related to collecting, cleaning, and preparing data for training the AI model.
- Training and Validation: Costs associated with training, fine-tuning, and validating the AI model.
- Integration: Expenses for integrating the AI system with the manufacturing process.
- Maintenance and Updates: Ongoing costs for system maintenance,

updates, and support.

$$Base\ Cost_i = (Tech\ Development_i + Data\ Collection_i + Training_i + Integration_i + Maintenance_i)$$

$$Cost_i = Base\ Cost_i + (Additional\ Cost_i \times Years_i)$$

- Base Cost - is the estimated initial cost of implementing AI-driven inspection for instance i.
- Additional Cost - represents additional annual costs for maintaining and updating the AI system.

This model incorporates an exponential growth component for revenue and a linear increase in costs over time. The equation combines both these aspects and accounts for the influence of market trends.