INTERNSHIP REPORT

INTERN NAME: Muhammad Huzaifa Bin Zafar
INTERNSHIP ORGANIZATION: National Electronics Complex of Pakistan (NECOP)
INTERNSHIP DURATION: 8 Weeks
MENTOR/SUPERVISOR: Sir Waqas

Table of Contents

1. Introduction:

The internship at the National Electronics Complex of Pakistan (NECOP) provided a comprehensive exposure to cutting-edge technologies in hardware trojan detection, cybersecurity, AI, image processing, deep learning, and reinforcement learning. My internship aimed to apply theoretical knowledge to practical problems, focusing on developing and refining skills in machine learning and computer vision.

2. Objectives of the Internship:

The primary objectives of this internship were:
- To gain hands-on experience in the fields of AI, machine learning, and cybersecurity.
- To work on real-life projects involving image processing, deep learning, and hardware trojan detection.
- To understand the practical applications of reinforcement learning in cybersecurity and AI.

3. Overview of the Organization:

National Electronics Complex of Pakistan (NECOP) is a leading organization in electronics, telecommunications, and cybersecurity. NECOP focuses on research and development in various domains, including hardware security, AI, machine learning, and embedded systems. The organization provides a robust platform for engineers and researchers to contribute to national-level projects with advanced technological solutions.

4. Learning and Tasks Assigned:

During the internship, I was assigned several tasks related to AI, machine learning, deep learning, and cybersecurity. I was involved in projects ranging from image processing and deep learning for defect detection in IC leads to hardware trojan detection and reinforcement learning applications.

5. Projects Undertaken:

5.1. Image Processing and Deep Learning Project:

The primary project I worked on during my internship was related to image processing and deep learning for detecting defects in IC leads. This project aimed to develop a deep learning model that could accurately identify various defects in IC pins using image processing techniques.

5.1.1. IC Lead Defect Detection:

The IC lead defect detection project involved analyzing IC pins to identify defects such as non-uniform color, tooling marks, exposed copper, bent or non-planar leads, excessive plating, missing pins, discoloration, scratches, oxidation, and solder issues.

The defect detection model was developed using Python with libraries such as OpenCV for image processing and TensorFlow for deep learning. The images were processed to extract features, and a deep learning model was trained to classify defects.

5.1.2. Techniques and Algorithms Used:

- Image Processing with OpenCV: OpenCV was used for pre-processing the images, including steps such as resizing, normalization, and augmentation. The images were scaled according to the actual lengths calculated from a microscopic microscope to ensure accurate detection.

- Deep Learning with TensorFlow and Keras: A Convolutional Neural Network (CNN) model was developed using TensorFlow and Keras. The model architecture included convolutional layers, batch normalization, dropout layers, and dense layers to enhance the model's ability to learn from complex patterns in the images.

- Data Augmentation: To improve the model's generalization capability, advanced data augmentation techniques such as random rotations, flips, brightness adjustments, and zooming were applied.

5.1.3. Data Processing and Augmentation:

The images were pre-processed and augmented to increase the dataset's size and variability. The augmentation techniques included:
- Random Rotation: To handle variations in IC orientations.
- Brightness Adjustment: To simulate different lighting conditions.
- Horizontal and Vertical Flips: To augment the data by flipping images.
- Zooming and Cropping: To focus on specific regions of interest in the IC pins.

5.1.4. Model Building and Training:

The deep learning model was built using a pre-trained VGG16 model with fine-tuning to leverage its powerful feature extraction capabilities. The model was trained on labeled data representing different defects. The training process included early stopping and learning rate reduction techniques to avoid overfitting.

The code snippets for data augmentation and model training are provided below:

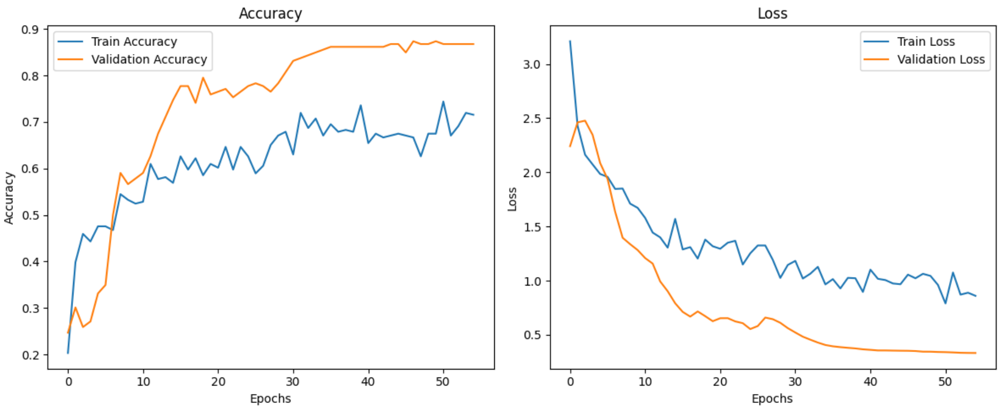
<u>data_augmentation.py:</u>

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

train_generator = datagen.flow_from_directory(
    'F:/training images/train',
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)
```

<u>pretrained_model.py:</u>

```
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Flatten, Dropout
```

```
from tensorflow.keras.optimizers import Adam

base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
x = base_model.output
x = Flatten()(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(11, activation='softmax')(x)   11 defects

model = Model(inputs=base_model.input, outputs=predictions)

for layer in base_model.layers:
    layer.trainable = False

model.compile(optimizer=Adam(learning_rate=1e-4), loss='categorical_crossentropy',
metrics=['accuracy'])

history = model.fit(train_generator, epochs=25, validation_data=validation_generator,
callbacks=[early_stopping, lr_reduce])
```

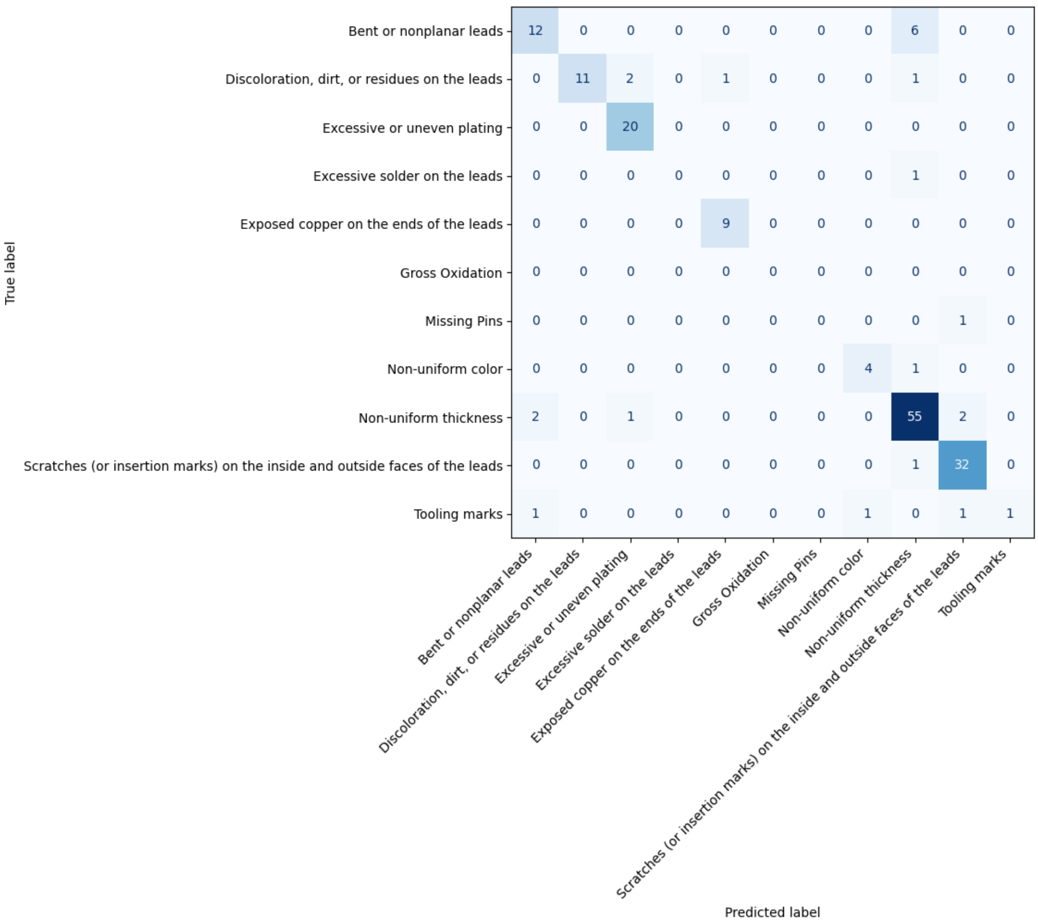


5.1.5. Results and Performance Evaluation:

The model's performance was evaluated using various metrics, including accuracy, precision, recall, and F1-score. A confusion matrix was generated to analyze the model's misclassifications. Below is the code snippet for plotting the confusion matrix:

confusion_matrix.py:

```
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

y_true = [actual_labels]
y_pred = [predicted_labels]

cm = confusion_matrix(y_true, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=class_names)
disp.plot(cmap=plt.cm.Blues)
plt.show()
```



5.1.6. Use of Microscope and Scanning Acoustic Microscope (SAM):

During the internship, a microscope was extensively used to visually inspect IC pins and leads for potential defects. The microscope enabled us to observe the IC pins closely to identify any defects that were not visible to the naked eye, such as non-uniform thickness, tooling marks, or exposed copper. This allowed for precise manual inspection and ensured that the training data for the deep learning model was accurate.

In addition to the optical microscope, a Scanning Acoustic Microscope (SAM) was utilized for non-destructive testing and internal inspection of ICs. The SAM uses focused sound waves to create detailed images of the internal structures of the ICs without causing any damage. This method is highly effective for detecting internal delaminations, voids, and cracks that may not be visible externally.

The Scanning Acoustic Tomography technique of SAM offers the following advantages:
- Non-Destructive Testing (NDT): It provides insights into the internal structures of components without dismantling or destroying them.
- Defect Detection: It can detect hidden defects such as cracks, voids, delamination, and other anomalies.
- High Resolution: The SAM provides high-resolution images of the internal features of ICs, making it an excellent tool for quality control and failure analysis.

By integrating data obtained from SAM and optical microscopy, a more comprehensive defect detection system was developed, enhancing the model's accuracy and robustness.

5.2. Hardware Trojan Detection:

This project focused on detecting hardware trojans embedded in integrated circuits. Various machine learning algorithms were employed to detect anomalies in hardware behavior that could indicate the presence of trojans.

5.3. Reinforcement Learning Application:

Reinforcement learning techniques were applied to optimize the detection process and minimize false positives in hardware security systems.

6. Skills Acquired:

During this internship, I acquired the following skills:
- Deep understanding of machine learning and deep learning models.
- Practical experience with TensorFlow, Keras, and OpenCV.
- Hands-on experience in data preprocessing, augmentation, and model evaluation.
- Knowledge of hardware security concepts, particularly in hardware trojan detection.
- Proficiency in using advanced microscopy techniques like SAM for non-destructive testing.

Certainly! I'll expand the "Challenges Faced" section by adding more detailed points on the challenges you encountered, such as having a smaller dataset, compatibility issues with libraries, and more explanations about the algorithms used.

7. Challenges Faced:

During the internship, several challenges were encountered that needed careful consideration and problem-solving:

- Limited Dataset Availability: One of the significant challenges faced was the limited availability of labeled datasets for defect detection in IC leads. A smaller dataset can lead to overfitting, where the model performs well on training data but fails to generalize to new, unseen data. This issue necessitated the use of data augmentation techniques to artificially increase the dataset size and diversity. Additionally, collecting more data and properly labeling it was time-consuming but essential to improve the model's accuracy.

- Algorithm Complexity and Selection: Choosing the right algorithms for the deep learning model was critical. While simpler models might not capture the intricate patterns required for defect detection, overly complex models can lead to overfitting. The selected model architecture, based on a pre-trained VGG16, required careful tuning to balance complexity and performance. The inclusion of additional layers such as Batch Normalization and Dropout was crucial to enhance the model's generalization ability and avoid overfitting.

- Library Compatibility Issues: During the development process, there were compatibility issues between different libraries. For instance, the version of NumPy installed was not compatible with the version required by TensorFlow, leading to runtime errors and warnings. This required troubleshooting, reinstalling, and testing multiple versions of libraries such as TensorFlow, NumPy, OpenCV, and Keras to ensure a stable development environment. Such incompatibilities slowed down the development process and required frequent adjustments.

- Hardware Limitations: Training deep learning models is computationally intensive and requires significant hardware resources, such as GPUs. Due to limited hardware availability, training times were longer, and model optimization had to be performed carefully. This limitation also restricted the use of more computationally expensive models and architectures that could potentially improve detection accuracy.

- Integrating Scanning Acoustic Microscope (SAM) and Optical Microscope Data: Combining data from both optical microscopy and SAM required careful consideration. The data from SAM provided valuable internal defect insights, while the optical microscope focused on external visual defects. Aligning and preprocessing these different types of data to work effectively together in the same model was challenging and required substantial effort to harmonize the datasets.

- Hyperparameter Tuning and Model Optimization: Fine-tuning the model's hyperparameters, such as the learning rate, batch size, and number of epochs, was a challenging and iterative process. Incorrect tuning could lead to either underfitting or overfitting. It required constant monitoring, evaluation, and adjustments to find the optimal set of hyperparameters that provided the best performance on both training and validation datasets.

- Handling Class Imbalance: The dataset contained images of different defects, but some classes were underrepresented compared to others. This class imbalance caused the model to be biased toward more prevalent classes. To address this, techniques like class weighting and oversampling of minority classes were employed. However, finding the right balance between these techniques required extensive experimentation and analysis.

- Integration of Multiple Defect Detection in Test Results: Accurately capturing and representing multiple defects in a single IC image was another challenge. The deep learning model needed to not only detect but also differentiate between multiple overlapping defects. This required designing the output results and evaluation metrics in such a way that they could handle multiple defect detections per image.

These challenges provided valuable learning experiences and opportunities for problem-solving and demonstrated the importance of flexibility, adaptability, and perseverance in a research and development environment.

8. Conclusion:

The internship at the National Electronics Complex of Pakistan (NECOP) provided a valuable opportunity to apply theoretical knowledge in a practical, real-world setting. The focus on IC lead defect detection through advanced image processing and deep learning techniques allowed for hands-on experience in developing and deploying machine learning models. This project encompassed various stages, from data collection using sophisticated equipment like Optical Microscopes and Scanning Acoustic Microscopes (SAM) to model development, training, and optimization.

The integration of these two forms of microscopy allowed for a comprehensive analysis of ICs, both externally and internally, without damaging the components. The project showcased the potential of non-destructive testing methods combined with deep learning to automate quality assurance processes in semiconductor manufacturing. The use of pre-trained models such as VGG16, combined with techniques like data augmentation, Batch Normalization, and early stopping, helped in overcoming challenges related to limited data and hardware constraints.

Moreover, the project involved significant problem-solving around data and library compatibility issues, hyperparameter tuning, and addressing class imbalance, all of which are common in real-world machine learning tasks. These challenges enriched the learning experience, underscoring the importance of flexibility, thorough understanding of algorithms, and continuous experimentation.

Overall, the internship was an excellent platform to develop skills in image processing, machine learning, and data analysis while working on a project that has direct industrial applications. It also highlighted the importance of interdisciplinary knowledge and collaboration in achieving effective solutions.

9. Recommendations:

Based on the experience and insights gained during this internship, several recommendations are proposed for future work and improvements:

1. Expand the Dataset and Improve Labeling Quality: A larger and more diverse dataset is essential for training more robust models. Efforts should be made to acquire more labeled data, possibly from different sources or under varied conditions, to ensure the model generalizes well to all types of IC defects. High-quality annotations and the inclusion of more defect categories could further enhance model performance.

2. Utilize Advanced Deep Learning Architectures: Future projects should consider employing more advanced architectures like EfficientNet, ResNet, or InceptionV3, which can capture more complex patterns in data. Transfer learning can be leveraged with these models to achieve better accuracy and efficiency without requiring enormous computational resources.

3. Invest in Hardware Resources: For deep learning tasks, especially those involving large datasets or complex models, access to high-performance hardware such as GPUs or TPUs is critical. Investing in these resources or using cloud-based solutions like Google Cloud or AWS could significantly reduce training times and allow for more extensive hyperparameter tuning.

4. Implement a More Comprehensive Defect Detection System: While the current model focuses on detecting external and internal defects, future systems should aim to integrate other inspection techniques, such as X-ray analysis or electrical testing data, to provide a more holistic defect detection approach. This multi-modal data fusion could lead to a more accurate and reliable automated quality assurance system.

5. Enhance Data Preprocessing and Augmentation Techniques: Although data augmentation was utilized to increase the dataset size artificially, more advanced techniques like Generative Adversarial Networks (GANs) could be employed to generate realistic synthetic data. Additionally, exploring more sophisticated image preprocessing techniques might improve defect visibility and feature extraction.

6. Refine Model Evaluation Metrics: The use of metrics like accuracy and confusion matrices was beneficial; however, more advanced evaluation methods such as F1 Score, Precision-Recall curves, and Receiver Operating Characteristic (ROC) curves should be considered to get a more nuanced understanding of the model's performance, especially in cases with imbalanced datasets.

7. Develop a User-Friendly Interface for Model Deployment: To deploy the defect detection model in a real-world manufacturing environment, a user-friendly interface with proper documentation and easy-to-understand visualization tools should be developed. This interface should provide clear insights into the defects detected and suggest possible corrective measures.

8. Focus on Model Interpretability and Explainability: For practical applications, especially in quality control and assurance, it is essential to understand the model's decision-making process. Tools like LIME (Local Interpretable Model-agnostic Explanations) or SHAP (SHapley Additive exPlanations) could be integrated to provide explanations for predictions, ensuring transparency and trust in automated systems.

9. Continuous Learning and Adaptation: The field of machine learning and deep learning is rapidly evolving. Continuous learning through courses, workshops, and staying updated with the latest research papers is crucial. Adapting to new technologies and approaches will keep the work at the cutting edge and ensure better performance and innovation.

10. Collaborate Across Disciplines: Working closely with experts in hardware, software, and domain-specific areas such as semiconductor manufacturing can lead to more effective solutions. Collaborative projects can enhance the understanding of practical challenges and improve the robustness of models developed.

By following these recommendations, future projects can build upon the foundations laid during this internship, leading to more advanced and effective solutions in the field of IC defect detection and quality assurance.