ADRS Software User Guide Version 1.0

ADRS Release 1.0

September 2019

Prepared by
Ddcmdatalabs, Chennai

Prepared for
IGCAR, Kalpakkam

ddl
Intelligence
Augmented

राष्ट्र की सेवा में परमाणु
इं गां प अ कें
IGCAR
ATOMS IN THE SERVICE OF THE NATION

Contents

CHAPTER - ONE

INTRODUCTION

This document describes ADRS application that is included in the delivery distribution. Each chapter describes the application that showcases specific functionality and provides instructions on how to use the application.

Documentation Roadmap

The following is a list of documents in suggested reading order:

- **Release Notes:** Provides release-specific information, limitations, known issues, next release and so on.
- **User's Guide:**
  - The software architecture and how to use it specifically in Microsoft Windows environment.
  - Platform Setup & Installation of ADRS Software
  - Describes how to use ADRS Software; to get users up and running quickly with the software.
  - Glossary

CHAPTER - TWO

ADRS Application Environment Setup

This chapter describes how to setup application environment. Application environment consists of

1. Development environment and
2. Runtime environment also known as Production environment

Though the setup for both the environments are similar, minor inclusions like U.I. build and ADRS Application usage augments the runtime environment. For this reason and also because of the project chronology, set up for these two environments are listed separately (2.1 and 2.10).

Development environment deals with creating software development environment and code snippets.

Production environment (for usage of ADRS) enables the user to classify RT images as and when new RT Images are available. Set up deals mostly with use of Command line for the production system.

2.1 Development Environment

2.1.1 Host Operating System

The application runs on Windows 10 Operating system.

Hardware platform is Intel i7 64 bits X-86 multi-core CPU with min 8GB memory

Graphic processor – (GPU) suggested is NVIDIA GPU with at least 2GB memory

2.1.2 Language used for Development

ADRS was developed using Python programming language under Windows 10

2.2 Installing Python 3.6

You can install Python from Download option found in the Python official site.

(Windows X86-64 executable installer from Python.org).

Easy option would be to use Anaconda to install Python and other libraries and create environment for development of ADRS

2.2.1 Installing Anaconda

Easier way of handling Python setup is to use Anaconda.

To install Anaconda:

1. Visit Anaconda.com/downloads
2. Select Windows O/s option
3. Download the *.exe* installer
4. Open and run the *.exe* installer
5. Open the **Anaconda Prompt** and run some Python code to verify the installation

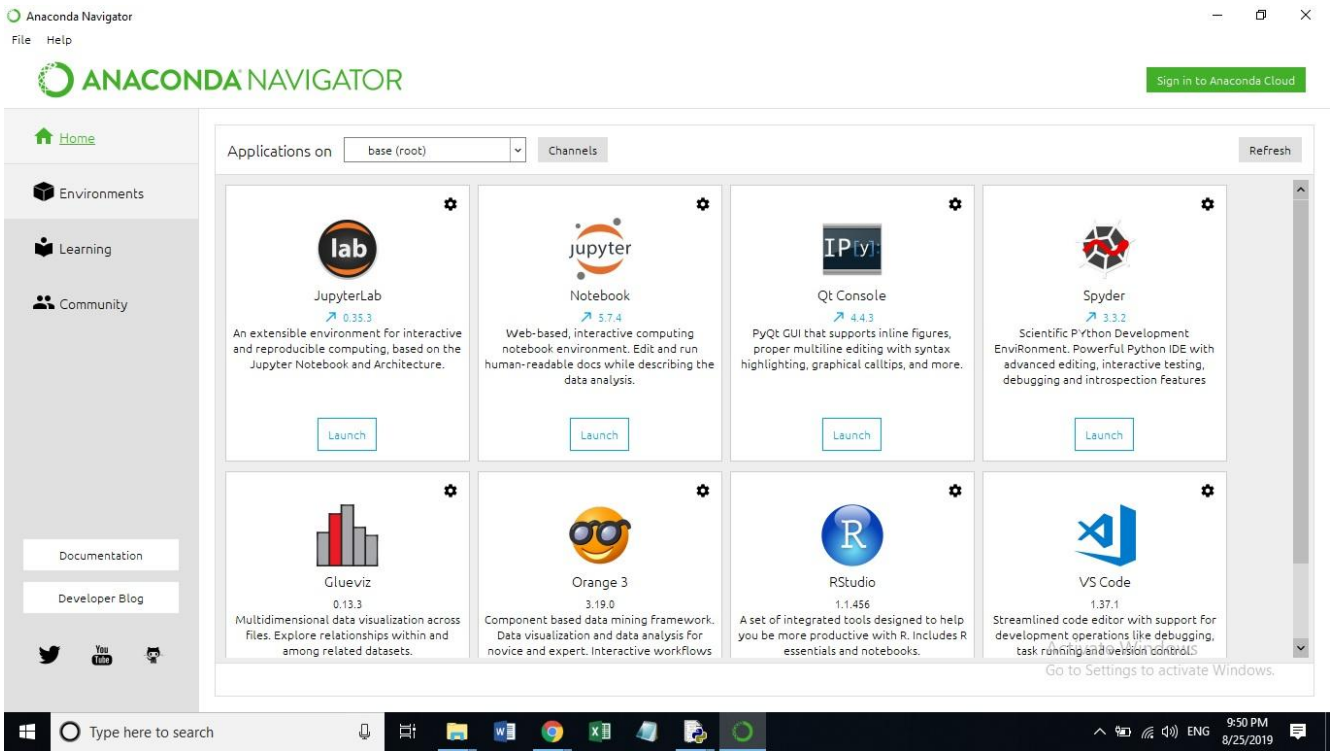
Use Navigator of Anaconda to install rest of Python and libraries like Numpy, Keras, TensorFlow (except Tensor Hub).

TensorFlow hub can be installed using command-line pip3 install.

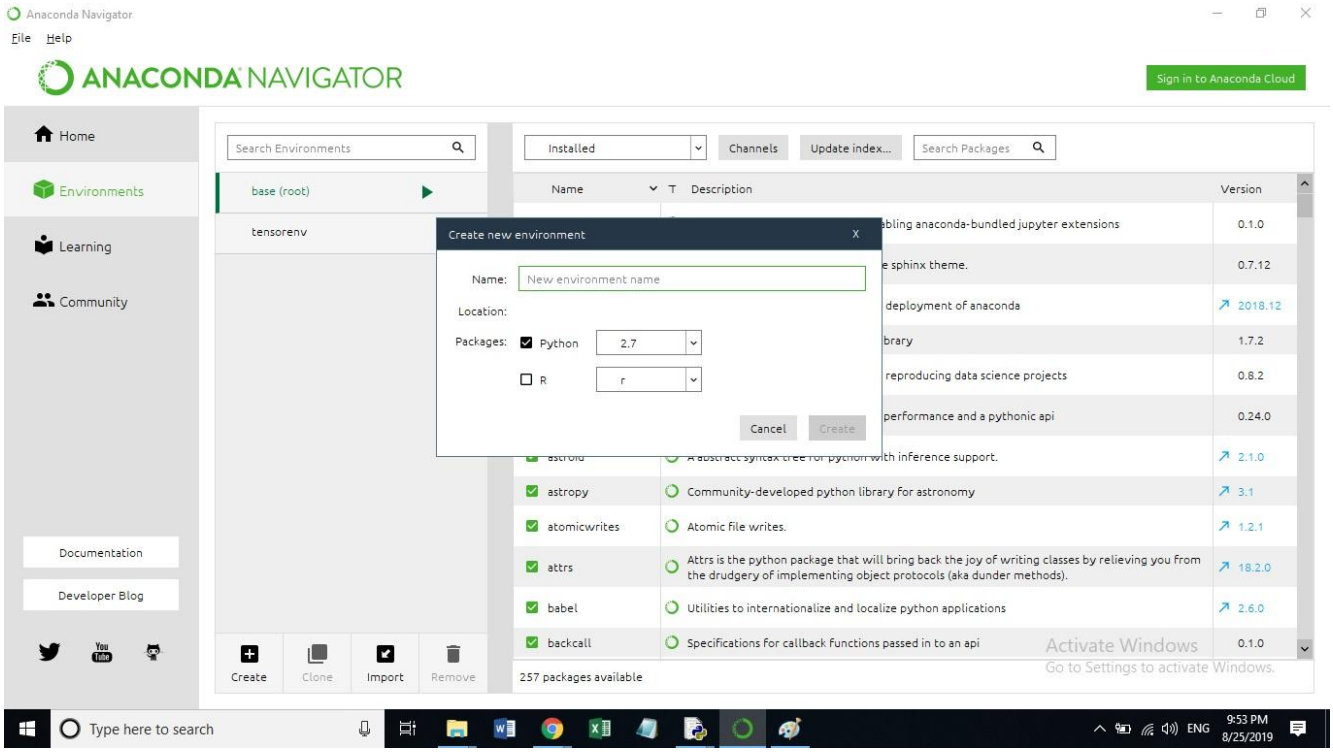
Anaconda open-source distribution is the easiest way to machine learning environment using Python on Windows. This can be achieved by using Anaconda navigator.

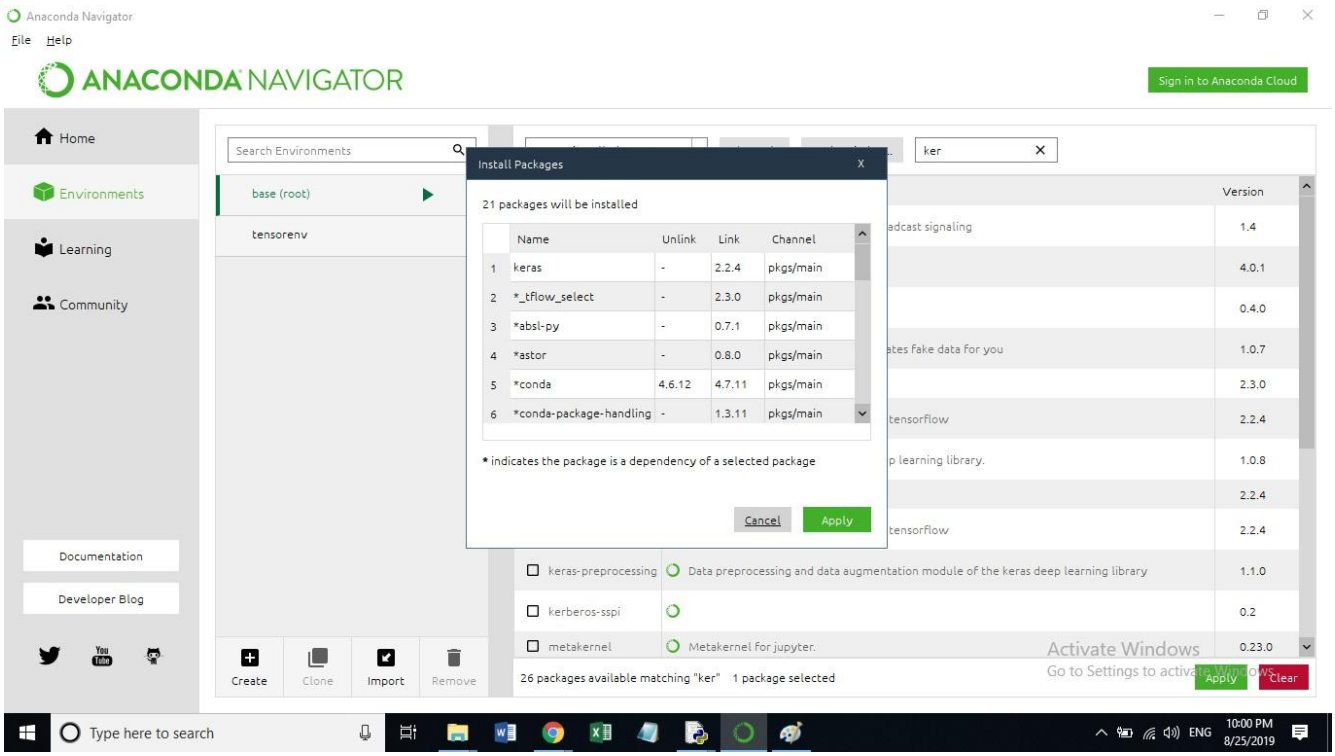
After the install is complete you can verify by clicking the shortcut 'Anaconda Navigator' from the recently added or by typing "Anaconda Navigator".

We need other packages like numpy, scikit, etc along with Python. By using Anaconda number packages like numpy, scikit, etc. are pre-installed with Anaconda.



2.2.2 Creating an Environment for the Project ADRS





2.3 Installing TensorFlow

ADRS uses TensorFlow as the deep learning framework. To install TensorFlow

Using Command-line:

Pip install tensorflow-gpu

This will install TensorFlow GPU in the home directory

Similarly you can install TensorFlow Hub. This contains TensorFlow graph and contains library for transfer learning.

2.4 Installing Keras

This can be done from command-line

>> pip install keras

2.5 Deep learning Model Setup

Inception V3 is the model used for machine learning in ADRS application.

2.6 Installing Inception V3

Inception V3 comes with keras.

To load pre-trained Inception V3 model,

```
From keras.application import inception_v3
```

2.7 Image processing Library Set up

Open source image processing library OpenCV 3.x is used for contour drawing and measurements.

2.8 Development and Runtime environment

Python 3.6.8 environment is used to develop and run the ADRS application code.

2.9 ADRS Application Setup & Runtime

As indicated earlier Python and Anaconda are already installed and ready to set up ADRS application.

2.10 Creation of ADRAS Application Environment

create a conda environment using the following command from the command line :

```
conda create -n <your_environment_name_here> python=3.6
```

For example: `conda create -n myenv python=3.6`

2.11 Activation of the Environment

```
conda activate <your_environment_name_here>
```

After activating the environment, we need to install the dependencies forADRS application.

Ensure that the dependencies are installed in the specified order.

2.12 Installation of TensorFlow and Dependencies

The following installs TensorFlow and all its Dependencies automatically.

```
conda install tensorflow
```

For TensorFlow with GPU support,

```
conda install tensorflow-gpu
```

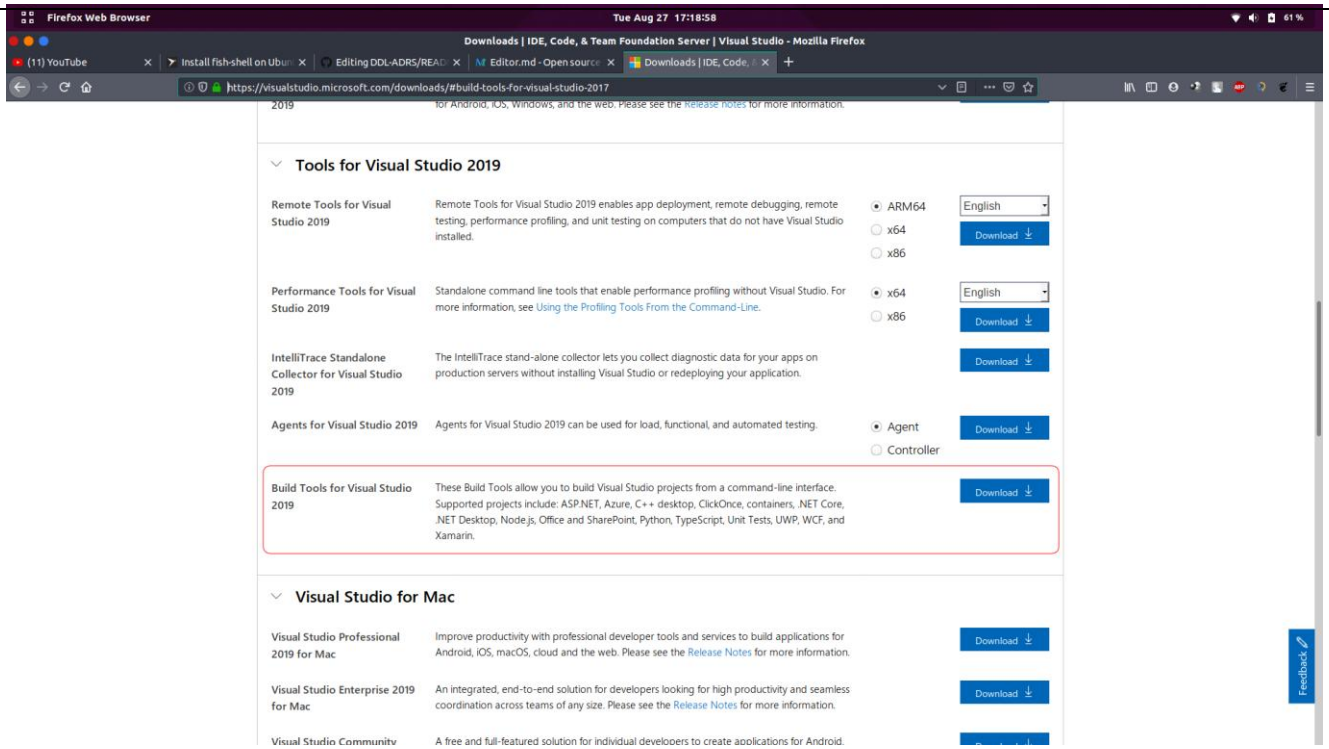
Then, install the following:

```
conda install keras
pip install opencv-python
pip install flask
pip install cython
```

Fianlly, we need to set up the yolo object detection model, used for weld detection.

But first, Visual C++ build tools are necessary. Those can be downloaded and installed from the following link.

https://www.visualstudio.microsoft.com/downloads/#build-tools-for-visual-studio-2017



Once build tools are installed, do the following:

1. Navigate to bin/darkflow
2. Open a command prompt window in that director
3. Activate the conda environment
4. Then run the following commands:

```
python setup.py build
```

```
python setup.py install
```

USAGE

After the setup is complete, navigate to the root directory in the command prompt and run:

```
python app.py
```

This will start the Flask server with the front end application.

STEPS TO INTEGRATE THE BACKEND

The image uploaded through the app will be uploaded to static/images. When the detect button is pressed in the app, the javascript file sends a request with the image url to accomodate varying filenames.

In app.py, locate the line

```
@app.route('/predict', methods = ['GET', 'POST'])
```

and do any image computations inside the `predict()` function. When reading an image through cv2, make sure to do it as follows:

```
cv2.imread(response.form['img_path'])
```

as `response.form['img_path']` corresponds to the image file.

After the image processing is done, please make sure to save the image to the "static/images" folder under a different filename. The "name" variable in the function corresponds to the filename. Considering that your final image is in the variable "img", save the image as follows:

```
cv2.imwrite(os.path.join(app.config['UPLOAD_FOLDER'],
f'{name}_final.png'), img)
```

After that, we need to send the image location back to javascript for it to display on the website. To do that, modify the dictionary called "response" as such:

```
response = "final_img_url":

app.config['UPLOAD_FOLDER'], f'{name}_final.png'),}
```

Feel free to add any other details, such as fault count, after the comma as a key value pair to the dictionary.

Now, go to main.js located in static/js. Here, navigate to the function `predictAndReturnImage()` and locate the line `success: function(response)`

Here, after the line `document.getElementById("predict-icon").style.visibility = "visible";`, add the following lines:

```
final_img_url = response.final_img_url;
$("#img2").attr("src", final_img_url);
document.getElementById("tab2-img").src = final_img_url
```

Now, the image should be displayed both on the main screen and on the image editor pane when it is clicked on.

CHAPTER - THREE

ADRS Application Architecture

3.1 Overview

Engineering industry uses Non-Destructive Evaluation methods to improve the quality of their implementations. One of techniques commonly used is Radiographic Testing.

3.2 The need for Non-Destructive Testing (NDT)

- Failure of a component could have very serious consequences
- To find critical defects in components used in high risk industries
- Commonly used in a variety of industries to improve product quality, reduce waste
- To improve productivity through reduced rework, the applications are nearly endless

NDT tools are becoming increasingly important as companies, large and small, strive to improve product quality and performance

3.3 Radiographic Testing

The process involves, taking X-Ray of joint (say welding) and look for defects in the X-Ray film before certifying the welding. This is manually time consuming and error prone. There is need to automate evaluation studies to track item testing. Use of digital imaging and computers in the evaluation of images is the first step in this direction.

3.4 ADRS as RT tool

ADRS is developed to automate defect recognition in the welding of joints using nondestructive evaluation method known as 'Radiographic Testing'. This test consists of using X-Ray beam to capture the defect such as pores, cracks etc., if any found in the welding. This is to ensure material integrity of the item under testing. The image of the welding is captured in an X-Ray film which is developed and viewed against an illuminator to examine defects if any. To view a large number of films is tedious and it is error prone to validate the welding captured in the films. Automating such a process is important when large number of welding are involved. The first step in automation is to digitize the X-Ray films so that digital images of the welding can be taken. This makes it not only easy to handle the images but also making the images amenable to machine learning as data input.

ADRS uses machine learning techniques of Artificial Intelligence. Machine learning is the science of programming a software application system so that the system learns from the data fed. Machine learning could be further divided into supervised leaning and unsupervised learning. Example of supervised learning is classification. Typically, the system is trained using data (input to the system) to classify new data instances.

- Film X-Ray is used for Radiographic Testing
- RT X-Ray films can be examined manually using an illuminator for the presence of defects
- Another way of examining RT films is using image viewing software which is very convenient to identify issues, defects or abnormalities relating to the component
- ADRS solution to RT method is aimed at automating classification of images already examined manually or using software and properly annotated as normal or defective
- ADRS uses machine learning techniques to automate classification of RT images

3.5 Challenges of machine learning

- Insufficient data
- Machine learning uses a lot of data for training the system.
- Even simple problems require thousands of sample data.
- Training data set should be representative of new cases we want to use.
- Poor quality of data
- Noise and image blur impacts quality of data making the system less likely to perform well.
- Overfitting the training data

  The system performs well with the training data but not on the new instances

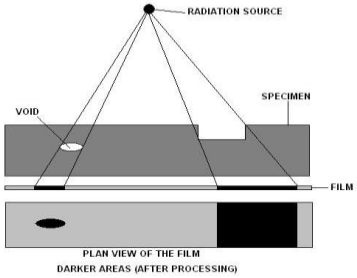
- Under fitting

  May not work even for training data because of oversimplification

3.6 ADRS Context Diagram

<table>
<tr><td colspan="2">RT film digitized images</td></tr>
<tr><td>Training of ADRS defect Classification</td><td>Using ADRS for defect classification</td></tr>
<tr><td>Image pre-processing to create input image dataset</td><td>New Image from RT repository</td></tr>
<tr><td>RT Image dataset for deep learning (DL)</td><td>Image pre-processing</td></tr>
<tr><td>Identification of DL model</td><td>Image classification</td></tr>
<tr><td>Training of the model</td><td></td></tr>
<tr><td>Image classification using trained model</td><td></td></tr>
<tr><td colspan="2">User Interface using Chrome on Windows 10</td></tr>
<tr><td colspan="2">Production System (ADRS)</td></tr>
</table>

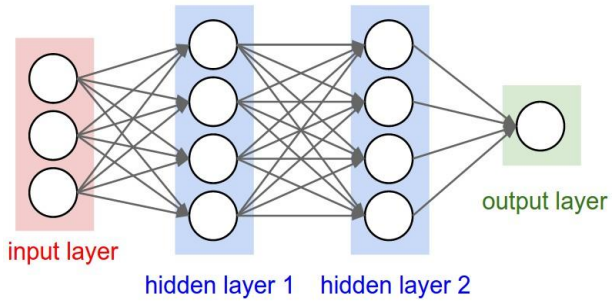
3.7 Application architecture



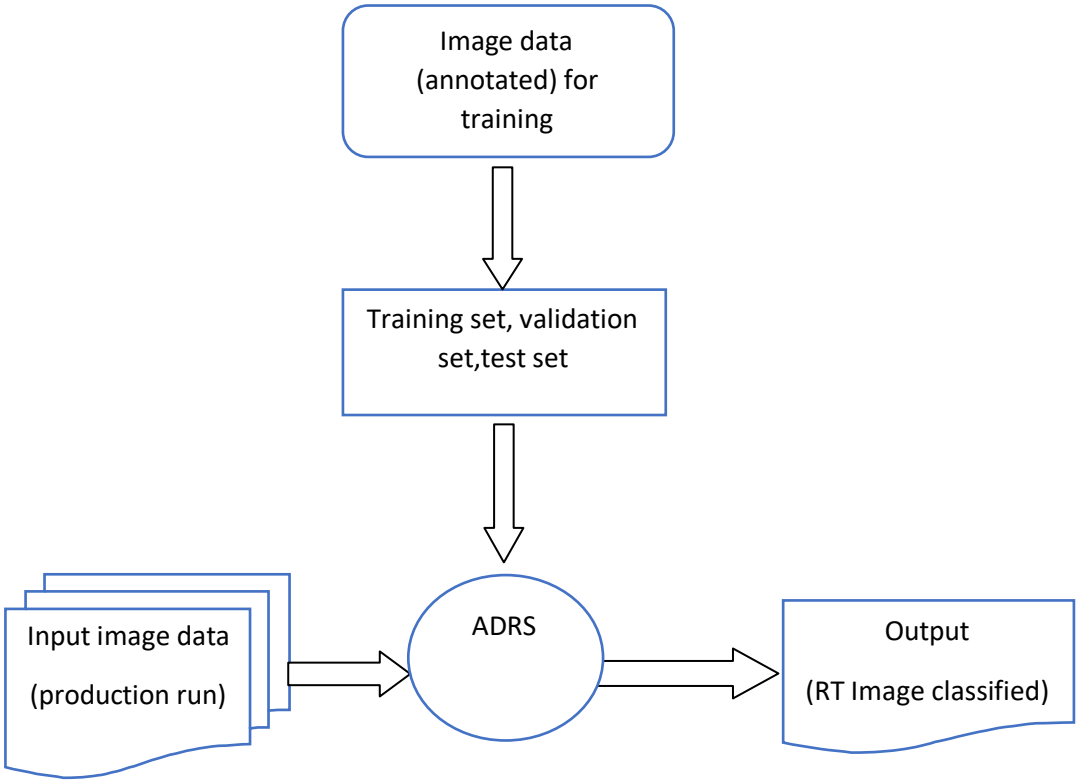
RT Film

Film digitizer (the images form the input data)

Machine learning System



3.8 ADRS machine learning - Schematic



3.9 Input data set

ADRS uses digitized images of Radiographic Testing (RT) Films. RT films are digitized using X Ray film digitizer.

This involves analogue film conversion to digital format using say 8 bits sampling.

These images are provided for machine learning and testing purposes. Number of sample images used: 1180.

3.10 Explanation

The following sections provide some explanation of the ADRS source code.

1. Classification of defects based on Deep Learning (machine learning) using Inception V3 Model with TensorFlow as the backend
2. Marking of defects using OpenCV
3. Defect metrics using OpenCV

3.11 Classification of Defects

The Inception V3 is a model trained using TensorFlow. This model is retrained in ADRS using the RT image input data given. The data goes through pre-processing step before Inception V3 is retrained with the preprocessed data. The trained model is used to classify designated sample RT Images as defective, normal, etc.

3.12 Image processing of the defective RT images

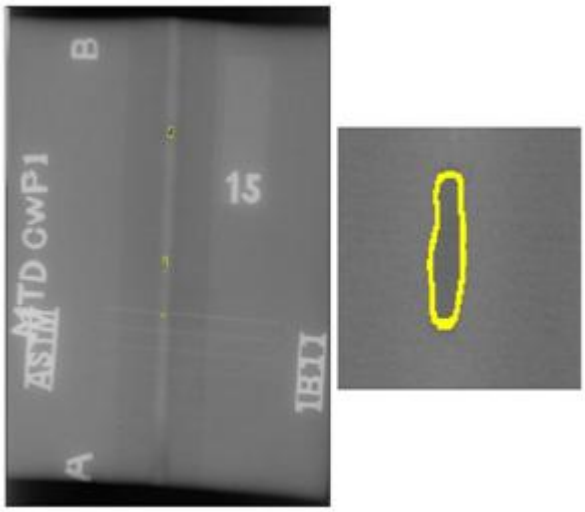
The image enhancement on the input data is done using contrast limited adaptive histogram equalization.

Canny edge detection is used on the enhanced image. Contour detection is done on the above image. Contours are marked. Perimeter of the contour is deduced (in pixels). Area of the contour is then calculated

3.13 Defect Metrics calculation

To get the measurement in in/mm, penetrometer meant for IQI is used. Template matching is used to identify penetrometer. Diameter of the penetrometer is calculated in pixels. Based on the penetrometer type table, the type and hence the factor for measurement is arrived at. This factor is used to compute the perimeter and area in in/mm

3.14 Defect Marking



3.15 Application Screenshots

3.15.1 Landing Page

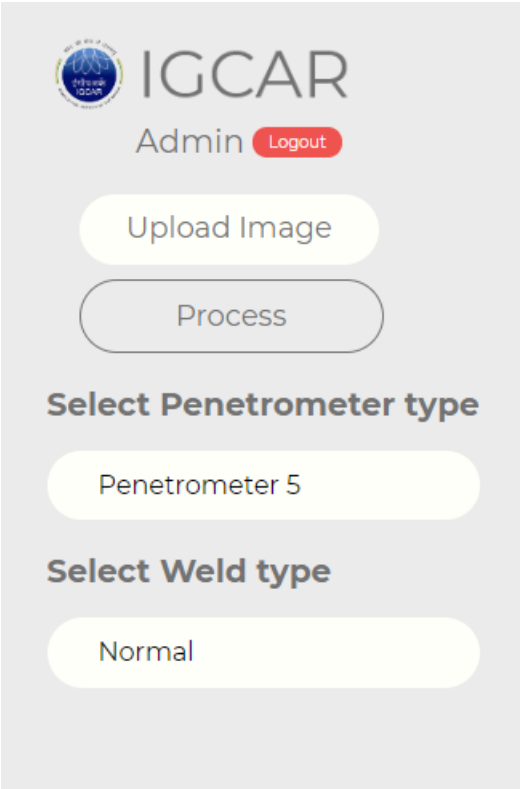
This page will serve as the landing page of the ADRS application, giving the user a small write-up of the ADRS system and navigate to the Login Page.

3.15.2 Login Screen

Enter the username and password to login to the ADRS system. The username and password will be shared separately.

Please note that in the Pilot run of ADRS, allows multiple users to login simultaneously, but only a single user id is made available.

3.15.3 Loading and Processing the image

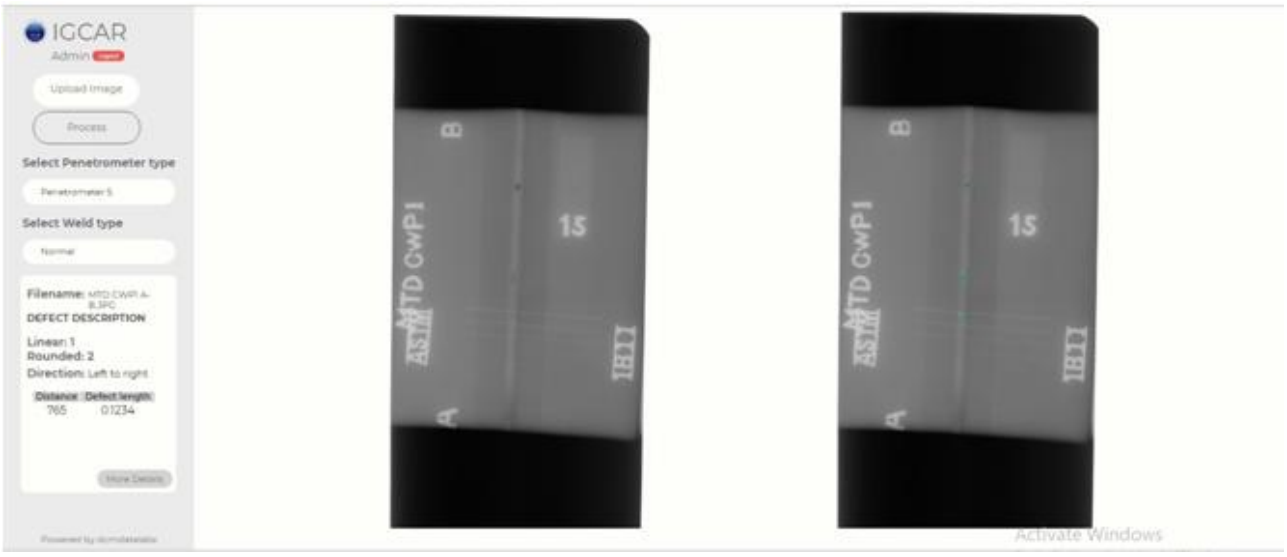


Use the *Upload Image* to select the image from the local storage.

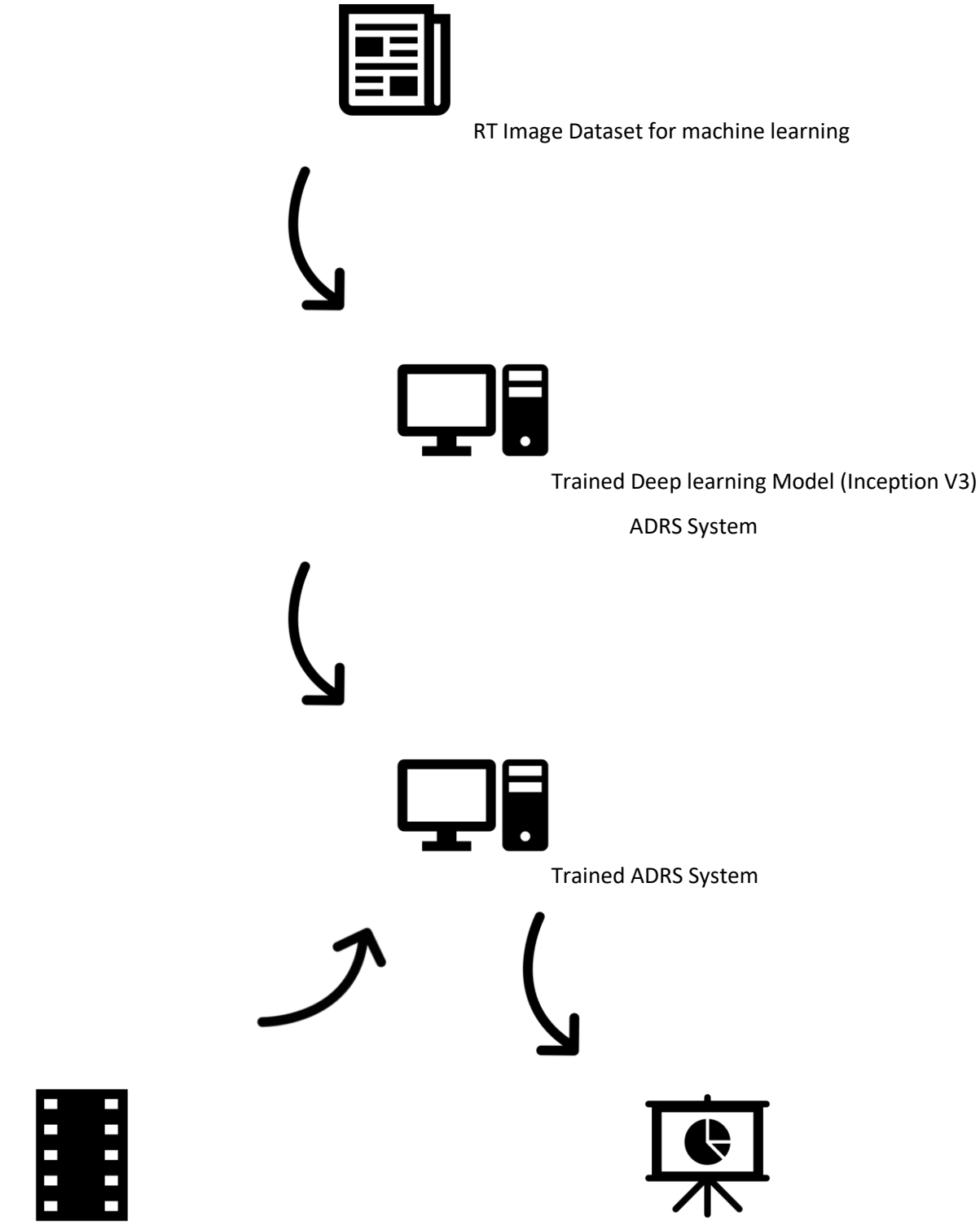
Once the image is selected use the *Process* button to begin the process.

3.15.4 Sample output display – defects marked

At the end of the process, both the original image and the processed image are displayed.



ADRS Usage Diagram



GLOSSARY

Artificial Intelligence (AI)

AI is the name coined for the use of intelligent computing systems for solving complex problems. It was proposed in 1950. Later rule-based systems (called expert systems) appeared. Only in the past ten years significant progress was achieved using different AI technique called 'Deep Learning' which used input dataset for classifying data instances. Progress in the powerful CPU and GPU made this transition possible.

Convnet

Core algorithm used in deep leaning is convolutional neural network also called convnets.

Deep Learning

It is a type of machine learning also goes by the name artificial neural network (ANN)

DICONDE

It is protocol standard for image storage and communication. It is based on medical DICOM image format. Image files are stored with image header detailing the image and body which is the image per se.

Edge Detection

Edge detection is an image processing method used to locate abrupt changes in intensity in an image.

Flask

Flask is a microframework for python web development

Graphic Processing Unit (GPU)

Advances in Deep learning started happening with the use of GPUs (developed for gaming devices). Now companies like Microsoft, Amazon provide GPU computing power for Deep Learning on the Cloud through Azure, AWS.

Histogram equalization:

The process of creating an output image by mapping each pixel in an input image based on histogram of the input image is called histogram equalization.

Inception V3

It is a convolutional neural network model used in image recognition

Keras

It is a high-level deep learning library works on top of TensorFlow, CNTK, Theano, etc.

Machine learning

Machine learning computer system is not explicitly programmed system but a trained system using input data.

NumPy

Is a set of mathematical functions library for handling multi-dimensional arrays and matrices

OpenCV

It is an open source computer vision and image processing library originally developed by Intel. Now available for Python programming environment.

Optical Density

Optical density of an X-Ray film is a measure of film darkening (transmitted density) when X-Ray passes through the film. It ranges from 0.5-4.0/4.5 describing 10,000 or more of intensity variation in the image.

Python

Python is a scientific programming language development environment in open source domain.

Template Matching

Locating a template image in any given image is called template matching which can result in 0 or multiple occurrences of the template image.

TensorFlow

It is an open source machine learning library used in the development of deep learning systems

TensorFlow Hub

It is a library for reusable machine learning modules.

X-Ray Film Digitization

Conversion of RT films to digital form is called digitization. Higher the OD of the film, higher contrast is possible. Useful digitizer can have a range say from 0 to 4 OD or more.

Release Notes

ADRS version 1.0 is the first release of the software focused on using deep learning paradigm and computer vision to create a prototype application to automate defect recognition in RT image and arriving at defect profile. Some of the limitations are ground up training of the system, database of image dataset conforming to DICONDE standards and standalone without link to RT Film digitizing system. Sensitivity improvement in defect detection and improvement in contour marking are contemplated in the future version in addition to addressing the above limitations.