Israel Mekonnen

Automated Aircraft Identification by Machine Vision

Helsinki Metropolia University of Applied Sciences

Degree **Bachelor of Engineering**

Degree Programme **Electronics**

Date **20.03.2017**

Author(s) Title Number of Pages Date	Israel Mekonnen Automated Aircraft Identification by Machine Vision 34 pages + 21 appendices 20.03.2017
Degree	Bachelor of Engineering
Degree Programme	Electronics
Instructor	Timo Kasurinen, Senior Lecturer

The main goal of this study was to develop an OCR solution for an automated identification of aircrafts at Helsinki-Malmi airport with the application of machine vision.

Cognex In-sight vision system was chosen as a platform for the system development. In-sight explorer was used in emulator mode in the absence of the vision camera. Patmax, OCRmax and Math tools were the key vision tools used. Images of aircrafts, taken at Helsinki-Malmi airport, were collected from different websites and used to train and test the vision tools.

The developed job was tested on 85 aircraft pictures, resulting in 84.7 % accurate reading. The result obtained is promising to continue working on Cognex platform, to develop a fully functional system that can read the aircraft register identifiers with 100% accuracy.

Keywords	Machine Vision, OCR, Pattern matching, Patmax, OCRmax

Contents

Abbreviations

OCR- Optical Character Recognition

ROI- region of interest

FOV – Field of view

RGB – red-green-blue

ARI - aircraft registration identifier

1 Introduction

Helsinki-Malmi airport is located in the district of Malmi in Helsinki. It is the second busiest airport in Finland. It is a hub of general aviation in the Helsinki region and home for several commercial pilot schools and aviation clubs. It is also a base of the Finnish Border Guard's Air Patrol.

Currently, air traffic control and aircraft registration in Helsinki-Malmi airport is done manually with the help of radio communication. Finavia, a corporation which operates and maintains the Helsinki-Malmi air traffic control, will no longer operate the airport since the beginning of 2017. This led for the need to find an alternative solution which handles registration of aircrafts before they take off.

The main focus of this final year project is to find a feasible solution for an automated aircraft identification of small aircrafts that intend to take off from Helsinki-Malmi airport with the application of machine vision.

In order to have a functional automated aircraft identification system, the system needs to be automated; meaning every process starting from initial step (Plane detection around the vicinity of the takeoff spot) to the last step (data communication) is handled by the system without any or minimal manual (human) interaction. As the system will operate near by the runaway of an airport, the system needs to be functional in various outdoor environmental conditions, such as day light changes and seasonal changes,

The general system design can be classified into three subsystems. The first subsystem is *detection and trigger* subsystem. It detects an aircraft which is at the takeoff spot and triggers the camera to capture an image which will be processed by the second subsystem. The second subsystem can be referred as *vision subsystem*. It analyzes the captured image to locate accurately the alphanumeric characters (aircraft registration identifies: ARI) on the body of the aircraft and read the characters correctly. The third subsystem can be referred as *data communication and data storage*. Its main function is to transfer the data extracted by the second subsystem to other operational and data storage systems.

Based on prior knowledge and experience in the field of automation, the cognex in-sight vision system was chosen by the project advisor Timo Kasurinen, Senior Lecturer at Metropolia University of applied science, as platform for the application development.

The application was developed using In-sight explorer in Emulator mode; an offline programming environment which allows to develop machine vision solution virtually without the presence of cognex cameras. Working on the first subsystem (detection and trigger subsystem) and the third subsystem (data communication and data storage) require the presence of a Cognex camera. As a result, the scope of this study is limited to the development of the second subsystem, which is the vision subsystem.

2 Theoretical Background

2.1 Machine Vision

Machine vision is the use of optical devices for noncontact sensing to automatically receive and interpret an image of a real scene in order to obtain information and/or control machines or processes. [1] It is the process of capturing and analysing an image for the inspection or control of various industrial and manufacturing processes.

Machine vision technology replaces or complements manual inspections and measurements with industrial cameras and image processing algorithms. It is used in various industries for various purposes, such as to automate production, increase production speed and yield, improve production quality, for microscopic inspection, closed-loop process control, robot guidance, precise non-contact measurement, assembly verification and for clean room environments and hazardous environments. The over all process of machine vision is shown in *Figure 1*.

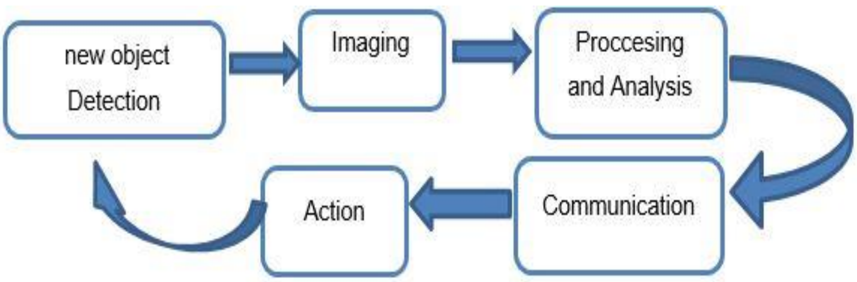


Figure 1: General process of Machine vision (adapted from [2]]

Each operations discussed in detail in subsequent sections.

2.1.1 Imaging (Image Acquisition)

Image acquisition is the first step in vision image processing. It is the process of acquiring an image using optical devices such as cameras or vision sensors. It is a crucial step because it highly affects the succeeding processes. The success of the image processing tools depends largely on the quality of the image input.

In digital imaging an image sensor, consisting of an integrated circuit with array of light detecting pixel sensors, is used to capture a digital image. The two technologies used for digital image sensors are CCD (Charge-Coupled Device) and CMOS (Complementary Metal Oxide Semiconductor). [2] Major components in image acquisition include camera, lens, lighting and Object.

Camera and Lens

All cameras used for machine vision are digital and they can be categorized as follows.

- *Vision Sensor System* – a vision system specialized for specific task.
- *Smart Camera* – a camera with built-in processor which is capable of image analysis and enable the camera to function standalone without a computer (PC).
- *PC-based System* – in this type of system the camera doesn't perform image analysis. It simply captures a picture and transfer it to a PC for image-analysis. Even though the camera doesn't do image analysis by itself, it is specifically designed for machine vision application. For Example, 3D cameras. [2]

The purpose of a *lens* is to focus the light beam that enters the camera to the pixel sensors in order to create a sharp image. Light beams which are not properly focused on the camera sensor create blurred image.

Angle of view and focal length are main differences that distinguish one lens from the other. The angle of view describes the angle range of the visual scene the camera can capture (*Figure 2*). Focal length is the distance between the lens and the focal point (imaging sensor).

Angle of view and focal length are related in such a way that a long focal length corresponds to a small angle of view and larger angle of view is related to shorter focal length.

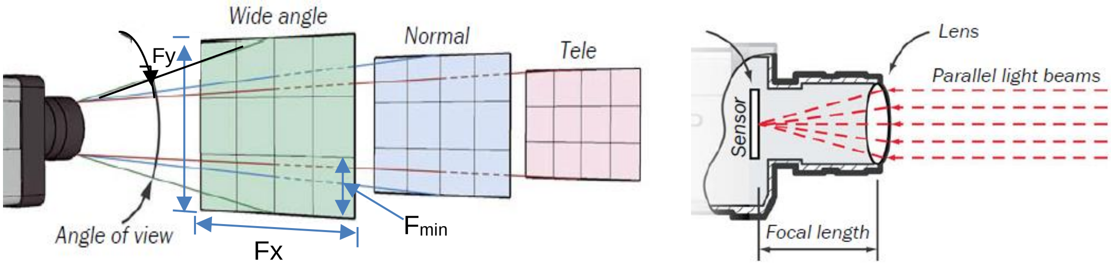


Figure 2: Angle of View (left) and Focal length (right) (adapted from [2])

Field of View(FOV) is the full area that the camera sees. It is specified by its width and height. *Working Distance (Object Distance)* is the distance between the lens of the camera and the object being captured. [2]

Pixels, Resolution and Intensity

The smallest building unit of a digital image is called a *pixel*. It is the smallest controllable and addressable element of a digital image. The pixel in the image corresponds directly to the physical pixel on the sensor.

Each pixels are arranged in two dimensional grid and has a coordinate address(x, y) as shown in *Figure 3*. A digital Image is a matrix (array) of pixel intensity values. The coordinate system usually used in image processing has its origin (0, 0)-coordinate at upper left corner of the image and its x and y coordinates take positive values. This representation corresponds to the matrix format which is very useful in image analysis operations.

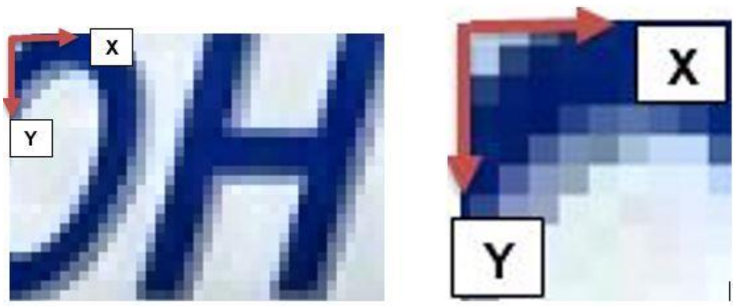


Figure 3: Coordinates based on two dimensional array of pixels

Resolution**:** Sensor *resolution* in 2D is expressed as number of pixel sensors in X-direction times number of pixel sensors in Y-direction. Image resolution describes horizontal number of pixels times vertical number of pixels. In other words it is the number of rows times number of columns of an array or a matrix that represent an image.

For machine vision application the required resolution can be calculated from spatial FOV dimensions and minimum pixel requirement to represent the smallest feature.

Let F - Spatial FOV dimension (Fx –horizontal dimension and Fy –Vertical dimension as shown on Figure-2)

F_{min} - The length of the smallest feature in FOV

N - Minimum number of pixels required to represent the smallest feature

D - Spatial length represented per one pixel

Rx - Horizontal resolution

Ry - Vertical resolution

The vertical and horizontal resolutions can be calculated as follows

$$D = F_{min}/N,$$

$$Rx=Fx/D, \quad (1)$$

$$Ry= Fy/D. \quad (2)$$

F_{min} is measured horizontally and vertically to calculate Rx and Ry respectively.

Grey scale describes monochrome brightness intensity of a pixel between black and white. *Intensity* is the numerical value of a pixel which describes its brightness. A grey scale is between 0 and 255, 0 being the darkest black, 255 being the brightest white and the intermediate values representing different intensity levels between the two extremes *(Figure 4*). 8 bit unsigned integer (1 byte) is used to store the intensity value one pixel. *Figure 7* shows a colour image converted to grey scale.

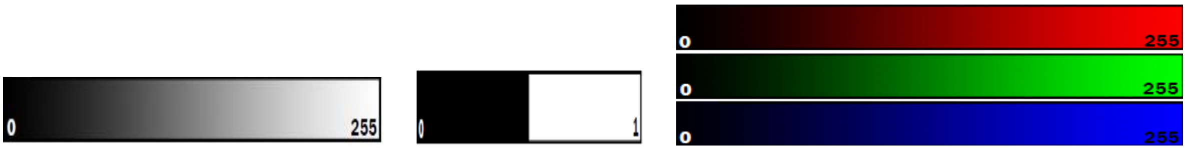


Figure 4: Grey scale (left), Binary image (middle), RGB Colour pixel components (Right)

Binary image is an image with only black and white pixels and no intermediate grey scale color (*Figure 4, Figure 6*). Only one bit per pixel is required to store the pixel value.

A pixel in a colour image has three components; red, green and blue (RGB). Similar to grey scale, each RGB components have intensities ranging from 0 to 255 (*Figure 4*). As a result, three bytes are needed to store full colour information of a pixel.

Contrast describes the relative difference between maximum and minimum pixel intensity values in given image.

Histogram shows the frequency distribution of pixel values in a given image. As shown on *Figure 5,* a histogram of a grey scale image is a continuous plot of grey scale values arranged in order of increase versus the frequency of their appearance. *Figure 6* shows a binary image and its histogram.

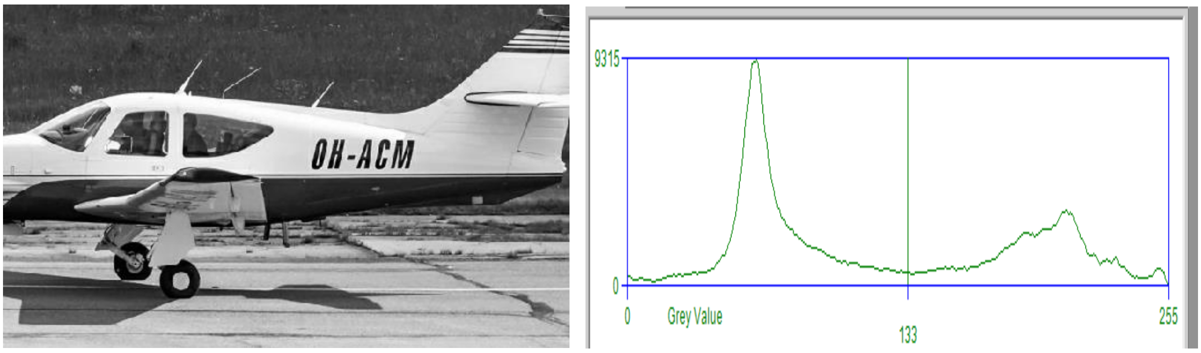


Figure 5: Grey scale Image (left) and its histogram representation(Right)

Figure 6 is obtained by binarizing *Figure 5* with a technique called thresholding.

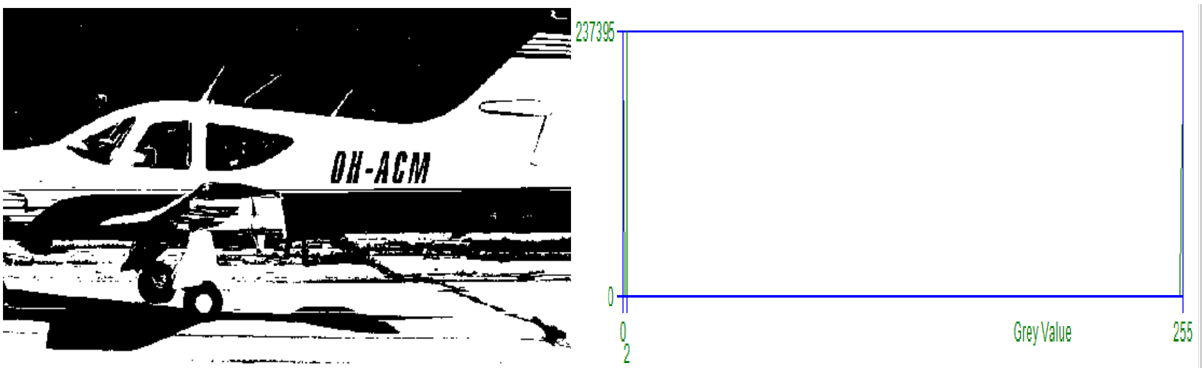


Figure 6: Binarized image (left) and its histogram representation (Right)

2.1.2 Processing and Analysis

Image processing and analysis is the step in which the image contents (mainly pixels) are analysed with image processing and vision tools (algorithms) to obtain the required information.

Region of Interest (ROI) is a specific area (part) in an image where the analysis is supposed to be done. ROI can be either static (when the location of object to be analysed is fixed in set of images) or dynamic (when the object location is not fixed throughout the set of the images to be analysed). *Figure 7* shows ROI for alpha numeric characters of the aircraft registration code.

Figure 7*:* ROI

Pixel counting is a method of finding number of pixels in a defined region that have intensities within a certain gray level interval[2]. Image is a two dimensional array of intensity values. Therefore, by applying pixel counting on these array of numbers length and areal measueremnts can be done pixel wise.

Filtering is image enhancemnt technique where pixels are operated based on some function of neighbourhood such as spatial operators in order to remove or enhance features. Image filtering is used for preprocessing before other vision tools are applied.

Thresholding is setting a minimum value to catagorize pixels so that vision operator operates on each catogory of pixels. It is a method of subdividing images directly into regions based on itensity values and it is one way of extracting objects from the background [3]. Thresholds can either be absolute or relative. In the context of gray scale images, an absolute threshold refers to a gray value (e.g. 0-255) and a relative threshold to a gray value difference, i.e. one gray value minus another.

Thresholding is applied frequently in binarization of gray scale images, where one absolute threshold divides the histogram into two intervals; below and above the threshold. All pixels below the threshold are made black and all pixels above the threshold are made white.[2]

Edge finding is image processing technique used for segmenting images based on local changes in intensity. It is method of finding a discontinuity on pixel values by operating on neghbourhood using gradient operators [3]. It is used to locate objects, find features and measure dimensions.

Blob is a set of contiguous(adjacent) pixels of the same color (intensity value) forming a distinct object. **Blob analysis** is finding and analysing these connected-pixel reagions in an image. It is used to identify and count objects, and make basic measuremnts of their charactestics [2].

2.1.3 Communication

At this stage, result obtained in prior step is sent to systems that use the result as input for further process. Different communication protocols such as TCP/IP, TCP/IP Modbus, File Transfer Protocol (FTP), SMTP, EtherNet/IP, MELSEC, PROFINET, POWERLINK, De-viceNet etc are used as means of communication in machine vision.

2.1.4 Action

This step is not part of the actual machine vision process. It’s a process carried out by a sys-tem looking for input from a machine vision system [2]. A robot adjusting its position based on the fixture input it receives from vision system, a product removal as a result of defect de-tected by a vision system, classifying products by reading their serial number are examples of actions depending on vision system input.

2.2 Optical Character Recognition (OCR)

Optical character recognition, or OCR, is a tool or an algorithm that reads and recognizes unknown text from images of scanned document, typed, scene-photo or hand-written. Each unknown letter is interpreted to known characters by comparing with a taught-in fonts.

Two types of OCR readers exist. One is the fixed font reader that uses fonts that are spe-cially designed for use with readers. The other is the flexible font reader that in principle can learn any set of alphanumeric characters. For robustness of the application, however, it is important to choose a font where the characters are as different as possible from one an-other. [2]

A typical OCR system comprises of the following steps shown in *Figure 8*. Like all other machine vision application, OCR stars with acquiring digital image. After acquiring a digital image, regions containing text are located and each symbols in the regions are isolated through a segmentation process. The extracted symbols are then pre-processed, in order to filter out non character segments. Then the extracted features are matched with fonts learned initially by the system. [4]

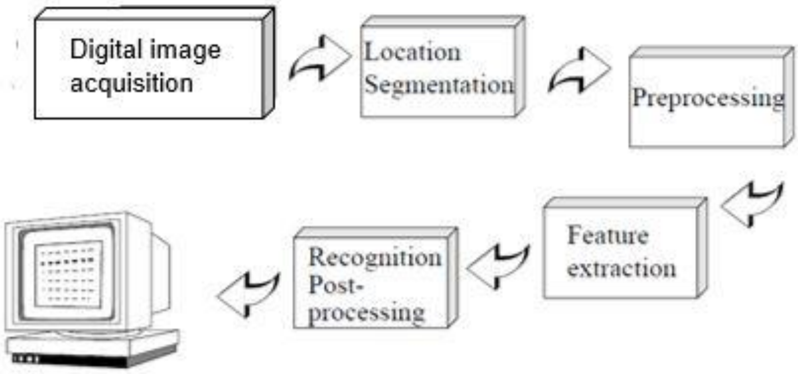


Figure 8: Components of an OCR-system (adapted from [4])

Reading a text from a body of an aircraft, which is ready for take-off, requires text extraction from a scene(natural) image which is far more complicated as compared to traditional OCR systems and machine vision ID applications, such as product label reading, serial number reading, barcode reading, where characters are typically monotone on fixed background.

The major sources of challenges in OCR from scene images are due to variations in background, lighting, foreground colour variation, foreground texture and font variations [5]. Bad quality imaging resulting in low resolution images and geometric distortions due to camera angle are also another setbacks [6]. In order to achieve better character recognition results, a scene image having text has be pre-processed to have a monochrome text and background where the background-to-text contrast should be high [7].

2.3 Pattern Recognition

Pattern recognition or pattern matching is a vision tool which looks for a pattern from a given image that matches a reference object or previously taught pattern in an image. Pattern matching can only be used when there is a reference object and the objects to inspect are (supposed to be) identical to the reference [2].

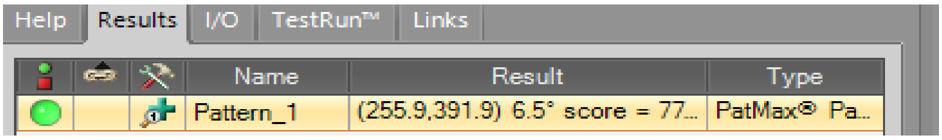
Pattern matching is used to locate objects, verify their shapes, and to align other inspection tools. The location of an object is defined with respect to a reference point that has a constant position relative to the reference object. [2]

Pattern matching tools typically give the location of the pattern in the image(x-y coordinates), angle (rotation), match score (% percentage) and number of similar patterns found.

The following figures show Cognex pattern tool (Patmax) set to find aircraft tyre: *Figure 9* is the reference pattern (model) to be matched, *Figure 10* shows the reference being matched on one of the aircraft images. The green colour around the tyre on the aircraft image shows the pattern found. *Figure 11* shows the result obtained from the pattern matching tool.

Figure 9: Reference pattern

Figure 10: Pattern matching on aircraft



			Name	Result	Type
			Pattern_1	(255.9,391.9) 6.5° score = 77...	PatMax® Pa...

Figure 11: Patmax result

2.4 Aircraft Registration

According to the Convention on International Civil Aviation, all civil aircraft must be registered using a unique alphanumeric string which is assigned by national aviation authority (NAA) of the country where the registration takes place. In this thesis, these alphanumeric strings are referred as aircraft registration identifier (**ARI**).

Every country has a prefix assigned to it and all aircrafts registered in a given country have a registration identifier that starts with the prefix assigned to the country. For example: the prefix assigned for Finland is ‘OH’ and hence all aircrafts whose registration identifier starts with ‘OH’ belong to Finland. D and N prefixes belong to Germany and United States respectively.

The alpha numeric characters placed after the prefix depend on the type of aircraft. For instance in Finland for lighter aircrafts the registration mark is formed of the nationality mark OH and a three-letter register mark. For helicopters the registration mark is formed using the nationality mark OH and a three-letter register mark beginning with the letter H. For home-built aeroplanes, helicopters and experimental aircraft the registration mark is formed of the nationality mark OH and a three-letter register mark beginning with the letter X. Similarly, for gliders and motor gliders: OH- followed by numbers, ultralight aircraft: OH-U followed by number and autogyros: OH-G followed by numbers. [8]

3 Cognex Vision System

Cognex is a company which produces machine vision tools for different purposes. The company’s main products include Data-man (for Barcode reading), Displacement Sensors-3D (for three dimensional inspection of a products), Vision-Pro and Cognex Vision Library (for determining products acceptability), checker (for inspection and part detection) and In-sight vision systems (for various inspections, identifications and guide parts).

Among Cognex machine vision products listed above, In-sight vision system has inspection tools (Pattern searching tools and OCR tools) closely matching the system requirements of the automated aircraft identification. As a result, the solution for the automated aircraft identification was developed with the application of in-sight vision development software called In-sight explorer.

In-sight vision system comprises family of vision cameras and software designed for inspections, identification and part guide. Major camera(sensor) products include in-sight 5600/5705 series, in-sight micro series, in-sight 7000 series, in-sight 2000 series and in-sight 5000 series.

3.1 Insight Explorer

Insight Explorer is an application which runs on a PC networked to In-sight camera. It is used to program In-sight camera. It uses spreadsheet as means of programming environment. It has two modes: Spreadsheet (which setup the spreadsheet itself) and Easy-Builder (a series of menu steps are used to create the spreadsheet). [9]

Because of its intuitive setups, easy graphical programming tools and its capability to make power vision tools, Easy-Builder was used to develop the *job* during the feasibility study. Job is an application (firmware that runs on the camera) developed by using In-sight Explorer tools.

Insight Emulator is an offline programming environment on in-sight explorer that can simulate various type of insight vision systems. The term offline programming is used to indicate the development of vision firmware in the absence/disconnection of the vision camera from the network.

Depending on the emulation mode selected, the availability of vision tools on insight explorer vary. For instance, color tools are available only if an emulator mode representing color vision cameras (sensors) is selected. Among the various machine vision tools provided by Insight-Explorer, only the tools required for the automated aircraft identification are discussed here.

3.2 Location Tools

Location tools are used to create a fixture which is used to locate a part in the acquired image quickly. This helps to create positional data which can then be used by other vision tools to define a reference point or ROI.

The location tools of insight explorer are able to locate a part in the image even if the part being inspected rotates or appears in different location in the image. In order to account for the location variation of the feature (part) three parameters are needed: (x, y) location and angle orientation. [9]

The three pattern matching tools of Insight explorer used as location tools are Pattern, *PatMax-pattern* and *PatMax Redline Pattern*. All of them report the X, Y coordinates, angle and score of the found pattern. The major difference among this tools is speed. The PatMax RedLine Patterns (1-10) and PatMax Patterns (1-10) location tools allow user to create a library of up to 10 different model patterns. Each model pattern must be trained using a separate image. [10]

3.3 Inspection Tools

3.3.1 OCRmax

The OCRMax (Read Text) identification tool is used to read and/or verify a text string within a region, after training and creating user-defined character fonts. [10]

The operation of the OCRMax function involves two phases: train-time and run-time. Train-time involves loading multiple images of the characters that will be read, extracting them from the image, segmenting them and creating a trained font database of characters. Run-time involves placing the In-Sight vision system online, acquiring images and extracting and classifying characters based on the trained font database.

The OCRMax function performs Optical Character Recognition through a process of segmentation and classification. Segmentation occurs first and uses threshold techniques to identify the areas of the image that appear to contain lines of text. After the text has been segmented into characters, the characters are trained and stored as a font database. Classification occurs during run-time, and is responsible for "reading" any text found after the function performs segmentation. This is done by comparing the images of the segmented characters to the trained characters in the font. [10]

Segmentation Process in OCRmax

The segmentation process has six steps: namely refine line, normalize, binarize, fragment, group and analyze [10]. All parameters mentioned below (indicated in italics) are explained in *Appendix 4, Table 3.*

In the first step, **Refine line**, OCRMax determines the location of the line of text within the ROI, and calculates the text's angle, skew and polarity. In finding the rotation and skew of the line of text, OCRmax relies on the orientation of the ROI, and the values set for *Angle Range* and *Skew Range* parameters. *Character polarity* is also another parameter which affects this step of the segmentation process.

Then in the **normalize** step, the region is normalized to remove unwanted noise before being binarized into foreground and background pixels. At this step, parameter settings for *Normalization Mode* and *Use Stroke width Filter* are taken in to consideration.

In the third step, **binerization**, a thresholding technique is applied to the normalized image based on the threshold value, in order to produce an image with only two greyscale values: Black (0) and White (255). Regardless of the *Character Polarity* parameter setting, text features are given a greyscale value of 255, and all of the background features are assigned a greyscale value of 0.

In the fourth step, **fragment**, within the binarized image, blob analysis is performed to produce character fragments, with each set of connected pixels considered as one fragment representing a single blob. This is done in order to determine whether the binarized text pixels in the third step are text pixels or not. Based on the analysis, some fragments are rejected and the remaining fragments are used to construct characters. Parameter settings of *Minimum Character Fragment Size*, *Character Fragment Contrast Threshold*, *Ignore Border Fragments* and *Maximum Fragment Distance to Mainline* are taken in to account to make precise blob analysis and fragment isolation.

At the fifth stage, **group**, character fragments are grouped together to form characters, and the characters are assigned a character region. The character region is a tight, non-editable bounding box enclosing all of the foreground pixels (characters) in the ROI. Optionally, additional analysis may be performed to determine more optimal groupings before forming the final characters. Four major operations are carried out under this stage, which are; merging two or more character fragments to form a character, splitting a character fragment or a group of character fragments into two or more (new) character fragments, trimming a character fragment either vertically or horizontally to make a smaller character fragment an discarding groups of character fragments. Parameters affecting this step include, *Character Fragment Merge Mode*, *Minimum Character Fragment Overlap*, *Minimum Inter-Character Gap*, *Maximum Intra-Character Gap*, *Minimum Character Size*, *Minimum Character Width, Minimum Character Height, Maximum Character Height*, *Maximum Character Width*, *Minimum Character Aspect Ratio* and *Character Width Type*.

The final step, **analyze**, is the continuation of the fifth stage in advanced mode. At this stage, fragments are grouped to form characters, based on *Global* information about all fragments within the ROI, rather than *Local* information, based on a few fragments. Parameter settings for *Segmentation Analysis Mode*, *Minimum Pitch, Character Pitch Type* and *Character Pitch Position* play important role for this step.

3.3.2 Math and Logic Tool

The Math & Logic Tool provides different functions to construct a formula which processes tool pass-fail data and job data, using standard math operators, logic, standard Boolean logic operators, statistics and trigonometry functions. The tool also allow to setup parameter transfers between the vision tools. Parameter transfer between vision tools can be between two PatMax tools, two OCRMax tools or PatMax and OCRMax tools.

4 Methodology

4.1 Collecting Aircraft Images, Editing and Preliminary Visual Inspection of Images

In order to develop and test the job on Emulator, 85 images of different aircrafts were collected from different websites. All the images were taken by normal camera at Helsinki-Malmi airport. Images were selected in such a way that they represent takeoff view where the FOV is perpendicular or nearly perpendicular to the direction of the flight so that the ARIs are clearly visible. The perpendicularity of the FOV reduces the geometric distortion of the characters coming from the angle of sight. Moreover, images representing different scenarios and taken at different weather conditions such as winter, night, daytime are included in the collection.

Note: All vision tool parameters mentioned in this chapter and other chapters are explained in *Appendix 4, Table 2 and 3.*

Resolution calculation and Image Resize

High resolution machine vision camera yield higher image quality but its price also gets higher compared to low resolution cameras. The following calculation is done in an attempt to find the minimum resolution required. The actual resolution of the vision camera can be decided after analyzing the trade-off with other factors such as cost and important specifications. Knowing the minimum resolution is sufficient for developing the job using In-Sight Explorer Emulator.

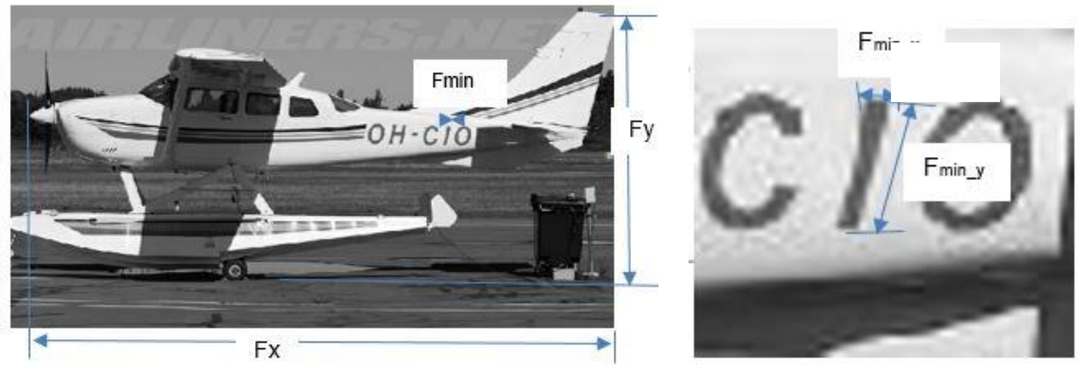


Figure 12: Spatial dimensions used to calculate resolution

In order to calculate the resolution, the dimensions shown on *Figure-12* are required. The dimensions are taken roughly by searching for exremum values from the collected pictures. Additional information for the following calculation can be found in chapter one.

Fy - the height of the biggest aircraft at Helsinki-Malmi airport

Fx - the length of the biggest aircraft (front to tail) at Helsinki-Malmi airport

Fmin= the width(Fmin_x) and height(Fmin_y) of the smallest character as printed on the body of the aircraft. ARI characters in aircraft images have the same height but different width. The narrowest character is 'I'. So Fmin is taken from the smallest **'I'** character. This dimension can also be measured by considering the narrowest character stroke.

D= spatial dimension per one pixel. It is calculated by considering Fmin.

Fx= 800 cm,   Fy= 400 cm,   Fmin_x= 4 cm,   Fmin_y=15 cm

Assuming D to be 1 cm per pixel, the 4 cm width of ‘I’ can be represented by 4 pixels, which was decided to be the minimum number of pixels to span a character stroke horizontally. Using equations 1 and 2,

$$R_x = F_x/D = 800\ \text{cm}/(1\ \text{cm/pixel}) = 800\ \text{pixels}$$

$$R_y = F_y/D = 400\ \text{cm}/(1\ \text{cm/pixel}) = 400\ \text{pixels}$$

Therefore, the minimum resolution required is 800X400. From the cognex camera specifications, the closest resolution for the calculated value is 800X600. Having the resolution calculated, In-sight 7200 which has resolution of 800X600 was choosen as a model for configuring Emulator.

Using the above spacial dimensions, working distance and the camera model, the focal length of the lens can calculated using lens advisor tool from Cognex website. The working distance (the maximum distance between the camera and aircraft at the take-off spot on the runway) was measured to be 25 meters.

Table 1: Lens focal length calcualtion

	Focal Length (cm):	Working Distance (cm):	Width (cm):	Height (cm):
Closest Higher	16.0	3018.868	800.000	600.000
Exact	13.3	2500.000	800.000	600.000
Closest Lower	12.5	2358.491	800.000	600.000

The lenses mentioned below are suggested by Cognex based on the above calculation.

LEC-63778 EDMUND 12MM 5MP Fixed Focus High Resolution

LFC-12.5F Fujinon 2/3" 12.5mm Lens F/1.4

LFC-16F1 Fujinon 2/3" 16mm Lens F/1.4

LFC-CF12.5 Fujinon 1" 12.5mm Lens F/1.4

The collected images had different pixel dimensions. In order to have images that have pixel size resemblance with In-sight 7200 resolution specification, resizing the images was needed. Furthermore, the resizing of the images is needed especially for images with dimensions larger than the camera specification. This is to avoid cropped view of the image on In-sight explorer. Microsoft picture manager was used to resize the pictures to 800 X 600, which is also the resolution specified on the data sheet of In-sight 7200.

Preliminary Inspection of Images

After collecting and resizing, initial visual inspection was done on the images. Some measurements such as, character height and word width were done using In-sight explorer; by navigating through axis values of points on the image. The purpose of this inspection was to study the variation in font style & font size, font location, possible background noises, differences and similarities in ARI. The findings of this initial study are included in the *Results* part. They were used to select the vision tools to be used from In-Sight explorer and set their parameters.

4.2 Emulator Configuration

This feasibility study was done using In-sight explorer in Emulator mode (detail explanation on Part 4). The selected camera model was In-sight 7200. Configuring Emulator requires accessing *offline programming key;* which can be found from Cognex website. Step by step guidance for Emulator configuration is provided on a lab manual submitted with this thesis paper. In-Sight Explorer window would not be active and tools are inaccessible unless Emulator is configured by inserting the *offline programming key*.

4.3 Problem Analysis, Vision Tools Selection and Vision Tool Setup

After configuring Emulator, the next steps were analyzing the result of the preliminary study, drafting a solution and selecting a vision tool to implement the proposed solution.

First step in every vision application is to find the part or feature of interest on which the vision tools operate. In this study, the feature of interest is the ARI on the body of the aircraft. Moreover, the first step in OCR application is defining ROI. ROI in this case is a region enclosing the ARIs (*Figure 9*). Unless the ARI is found in the same location in every image, the location (ROI) has to be configured with location tools. As shown on *Appendix 1* the ARI's location is dynamic and varies from image to image. Even within the aircraft itself, the ARI is positioned on different locations.

Besides the location variation, there are other texts on the body of the aircrafts (For instance, such as promotions, websites, company names and logos) which are not part of the ARI (*Appendix 2).* The ROI has to be fixed in such a way that it excludes these texts and text looking features. As a result, **Patmax**, the latest and powerful pattern matching tool supported by In-sight 7200, was selected as location tool.

After defining the ROI using pattern matching tool (Patmax), the next steps are image enhancement operation over the ROI and reading the characters inside the ROI. For both processes **OCRmax**, the latest cognex character recognition and verification tool, was chosen.

The **Math and Logic** tools were also needed in order to enhance interdependent communication between the selected tools, to turn the tools on & off in different conditions and to make logical decisions based on the results obtained from the tools.

Selected Vision Tools Setup

In this section, first the final solution is presented as shown on *Figure 13*. Then the problem analysis and steps followed in drafting the solution are discussed. The screen shot on Figure 13 was taken from In-sight explorer. After analyzing various efficient ways of configuring In-Sight vision tools, three Patmax tools, three OCRmax tools and four Math tools were added. The action flow between the added tools is further elaborated using flow chart diagram *(Figure 21)*.

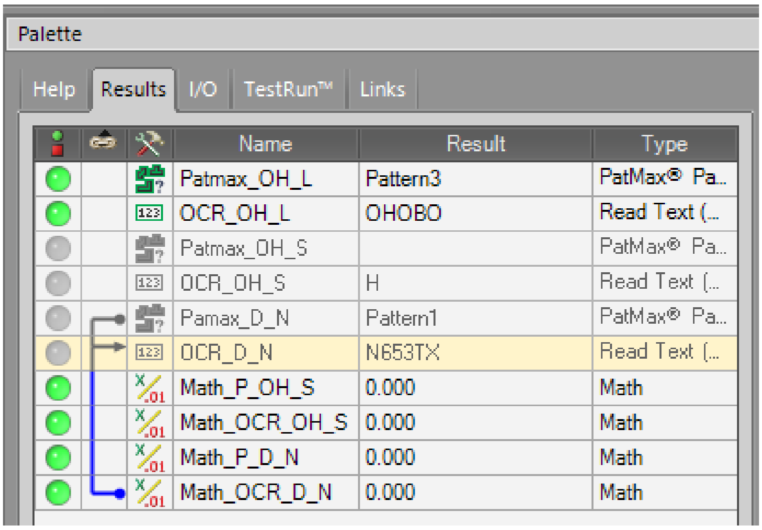


Figure 13: Vision tools added for job development

In setting up Patmax, the first step is to identify the feature that is common in all or most images . Eventhough the feature is not supposed to have fixed location in all images, it is required to have fixed location relative to the ARIs so that it can be used by Patmax to accurately locate ROI (ARI) for OCRmax.

At the beginning, different parts of an aircraft such as wing, tail, wheel and fuselage of the aircraft were taken into consideration to select common feature. However, because of the differences in aircrafts' type, size and shape, none of them could be considered as common patterns that could appear in every image.

The next seemingly common feature taken in to consideration was the prefix in ARI. As explained in Chapter 1, all aircrafts registered in a given country have a registration identifier that starts with a common prefix assigned to the country. As shown on the preliminary study result, mainly three prefixes (OH-, D- and N) that are common among ARIs are identified. As a result, this prefixes could be used as common model features for Patmax.

There are also prominent differences among prefixes as explained in the preliminary study results in chapter 5. Major differences are due to prefix characters, font style, font size, polarity and skew. Having these five major variations, the number of patterns to cover all possible Patmax model features could be calculated from the following tree.

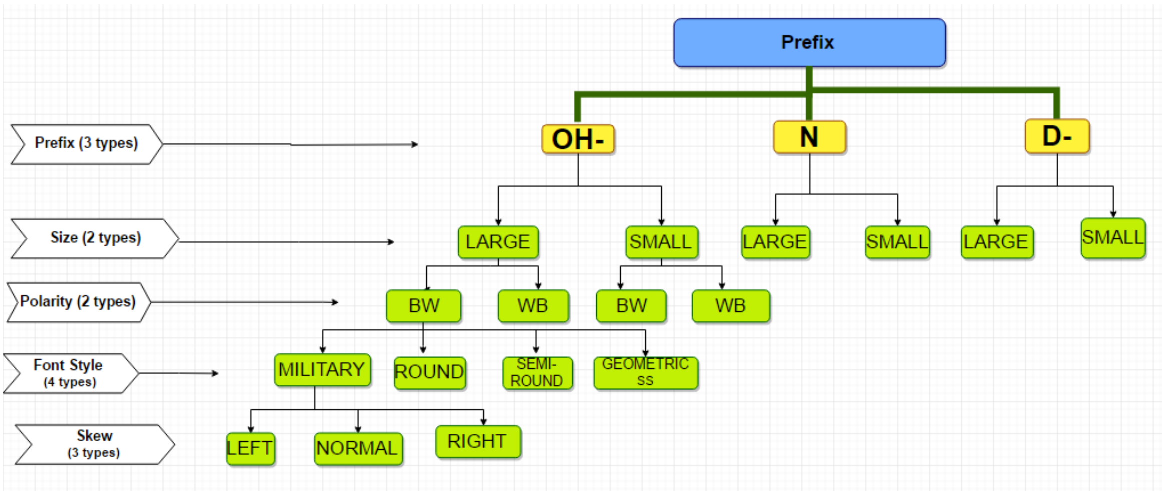


Figure 14: Partial view of possible pattern models of ARI prefixes

One Patmax Pattern (1-10) tool can take up to ten different *model patterns,* which provides a possibility of accommodating different font prefixes (**OH-, D-, N**). Besides, for some of these differences Patmax provides tolerance ranges through its parameters.

Completing the tree results 144 (3x2x2x3x4) model patterns which require 15 Patmax tools. This creates a *job* that demands huge memory. Consequently, the number of pattern models has to be reduced.

The first solution for the above problem is finding the font styles that result approximate scores and representing them by one prefix that has high score. With small test done on couple of pictures using In-sight explorer Round, Semi-round and Military prefixes have close scores and can be best represented my military font. The left and right skew prefixes of Military fonts also have closer scores with the left and right skew of the Geometric-sans-sarif font. Besides, from the collected images there are only four aircrafts starting with N and D prefixes with all being large fonts. As a result, the tree rooting from **N** and **D**- can be modified to have only four pattern model. The minimized pattern models can be seen from *Figure 15*.

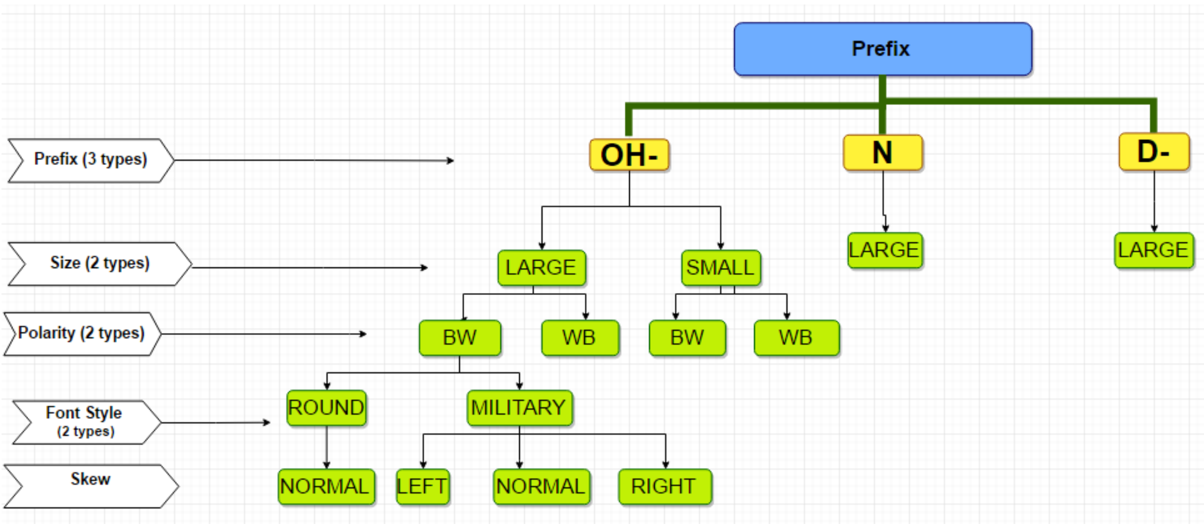


Figure 15: Minimized pattern models of ARI prefixes

Scale tolerance parameter of Patmax covers only ±50% of the actual pattern-model size. If the average of the maximum (35 pixels) and minimum (8 Pixels) size of the prefixes (by character height) is taken, which is approximately 22 pixels, with the scale tolerance (±50%) it covers only the range 11-33 pixels. Consequently, the prefixes have to be divided into two (large and small) according to the font size.

Patmax doesn’t compensate for skew angle. Including slant cases (right and left skews) improves the accuracy of fixture angle calculated by PatMax.

After testing the proposed model patterns (prefixes) and considering variations explained above, the model patterns were reduced from 144 to 20 and arranged as shown on Figure 16, 17 and 18 for the convenience of OCRmax.

Fiure 16: Model features for first Patmax tool (Patmax_OH_L)

Figure 17: Model features for second Patmax tool (Patmax_OH_S)

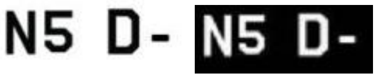


Figure 18: Model features for the third Patmax tool (Patmax_D_N)

Note: Names of the OCRmax, Patmax tools and Math & Logic tools (OCR_OH_L, Patmax_OH_L, Math_P_OH_S and other tool names) were given by the author while working on In-Sight explorer. Any name can be given to the tools.

The first Patmax tool (Patmax_OH_L) covers **OH-** pattern-models with large font size, both polarities (BW and WB) and both skews (right & left skews). The Second Patmax tool (Patmax_OH_S) covers the same pattern-models as the first one but with smaller font sizes. The third Patmax tool (Patmax_D_N) covers the pattern-models starting with prefix **N** & **D-** and both polarities (BW and WB)**.** From the collected images only four images, having the specified prefixes, correspond to this Patmax tool.

Unlike the general approach for the first two Patmax tools, the models for third Patmax tool were minimized to cover only the four aircraft images in the collection. Depending on additional variations, more models could be added. Since Patmax can accommodate ten models and only four models were used for Patmax_D_N, the rest of the six model spaces were used to train models that start with **OH-** but could not be found by Patmax_OH_L and Patmax_OH_S**.**

After defining the model region for Patmax, the next step is configuring its parameters based on measurements made and repetitive tastes made in order to find the best values and settings. The parameter configurations for the three PatMax tools are elaborated in *Appendix 4, Table 5*.

Completing Patmax tool configuration, next step was adding and configuring OCRmax tools. As shown in figure 13, three OCRmax tools were added. Besides the compelling reason discussed above (large and small font size coverage by Patmax), the area of the text region in a given OCRMax tool is fixed. As a result, developing a job with one OCRMax operating on all images with the same parameter configuration, creates less accurate or completely inaccurate reading where in some images characters are excluded while on the other images unnecessary background noises are included. Adding separate OCRmax tools depending on the font size enables to define different ROIs for the OCR having areal dimensions that are proportional with the character sizes. By using smaller ROI area for small fonts, the noises enclosed by the region can be minimized.

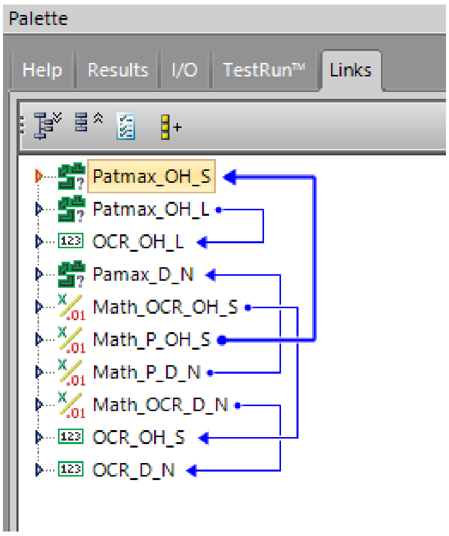


Figure 19: Added vision tools the link between them

OCR_OH_L corresponds to Patmax_OH_L. It is linked with the Pass/Fail result of Patmax_OH_L; meaning it is enabled only when Patmax_OH_L pass (succeed in finding the trained part). In addition, it gets *fixture* argument (angle and coordinate values of the found pattern) as an input from Patmax_OH_L, which then is used to locate ROI accurately. Like-wise, OCR_OH_S and OCR_D_N correspond to Patmax_OH_S and Patmax_D_N respectively and they rely on Pass/Fail and fixture arguments of their corresponding Patmax tools. *Figure 19* shows how the PatMax and OCRMax tools are linked.

Another essential step in setting up OCRMax tools is Font training. By the very core principle of machine vision, which is machine learning, OCRMax must be trained to learn the fonts it is intended to read. Due to the ARIs font style variation, the OCRMax has to be trained with multiple instances for each characters.

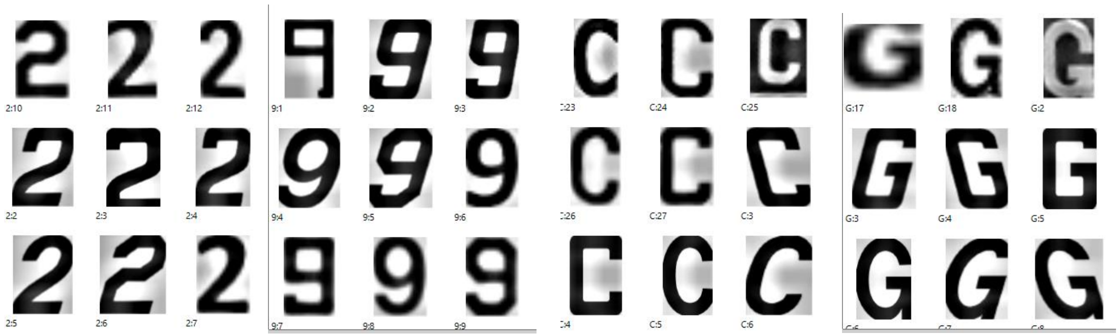


Figure 20: Trained fonts

OCRMax has built in font data base. However, the data base does not have sufficient instances of fonts. So, font training with additional font character was needed. In order to do so, image templates consisting of additional fonts that have closer resemblance with the ARIs were prepared (Appendix 5). In addition to the image templates, some aircraft images were also used for the font training. *Figure 20* shows sample fonts taken from the trained font data base.

OCRMax provides two ways of font training: *Auto-Tune Dialog (*a method which combines the segmentation and training phases into one step and calculates the optimal segmentation settings automatically) and *Manual Segmentation* (where parameters are adjusted manually until the text is correctly enclosed within individual character regions).[10] Both methods were used to build the font data base. The step by step procedure of font training is explained in the lab manual.

After training fonts and building the font database, the next step was configuring the *setting, segmentation, advanced, space and variable length parameters* (*Appendix 4*) for each OC-

RMax tool based on measurements and repetitive tastes made in order to find the best parameter values that works in all or most aircraft images. The parameter definitions and configurations for each OCRMax tool are explained in Appendix 3 and Appendix 4 respectively.

The ARI characters can uniquely identify the aircrafts without the hyphen that is placed in between the prefixes and the rest of characters. By ignoring the hyphen while defining parameters that are related with character size, smaller noise fragments that have size close to hyphen can be filtered out. As a result, in the font training step hyphen was neglected.

Having completed the OCRMax configuration, the next tool to setup was the Math and Logic tool. In this job development two ways were used to pass an argument (parameter value) between vision tools. The first way was by creating a direct link between inputs and outputs of the vision tools. *Figure 21* shows screen shot taken from the link tab. Cognex vision tools have four input/output data types: Floating, Integer and String and. On Easy builder, if a given vision tool need to receive a single output argument from another vision tool, a direct input-output link can be created graphically on the links tab. For more step wise explanation the lab manual can be referred.

The second method applies in a situation where an input parameter of a vision tool depends on the combination of two or more than two output parameter values. In this case, the Math tool was used to combine the output parameter values by logic and math operators and then its result was linked graphically to the input parameter of the corresponding vision tool. *Figure 21* shows a partial view of the links created.

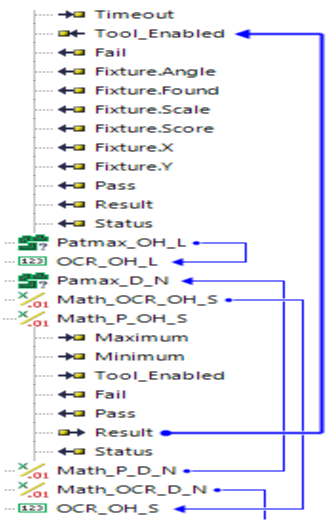


Figure 19: Partial view of the link between the tools functions

The action flow of the developed job is shown on the flow chart (Figure 15). First an aircraft image is processed by Patmax_OH_L to find the trained models.

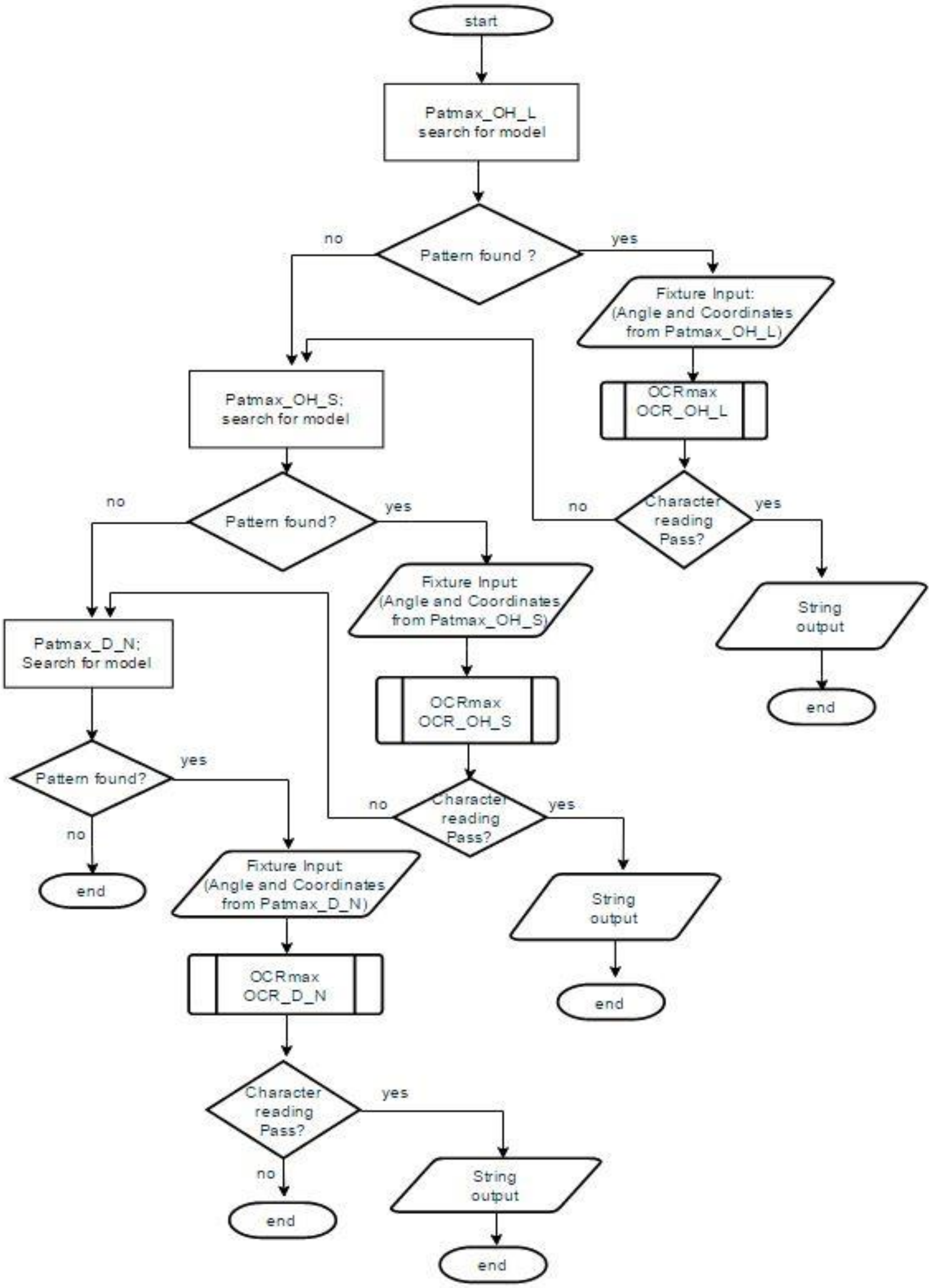


Figure 21: Flow chart showing the action flow

The *Tool Enable* parameter of OCR_OH_L is linked to the Patmax_OH_L parameter storing the *Pass* result. If the trained model is found, OCR_OH_L gets turned on and receive a *fixture* as an input from Patmax_OH_L. Then the OCR_OH_L carries out the segmentation and classification process in the ROI positioned based on the fixture received (details of the OCRMax process explained in chapter 3). If OCR_OH_L pass (characters are read correctly) the process ends and characters are stored in *String* parameter.

The conditions in which an OCRMax tool fails were defined as,

- If it fails to segment and classify the characters enclosed in the ROI (result can be accessed from either Fail or Pass parameters).
- If the characters read are less than five (the minimum number of ARI characters) or greater than six (the maximum number of ARI characters).

Patmax_OH_S is turned on based two conditions: If Patmax_OH_L fails or if OCR_OH_L fails. Therefore the outputs of these two tools are combined by Math_P_OH_S as shown on Lising 1.

```
Patmax_OH_L.Fail || (Patmax_OH_L.Pass && (OCR_OH_L.Fail ||
OCR_OH_L.Result_Length<5 || OCR_OH_L.Result_Length>6))
```

Listing 1. The code for Math_P_OH_S

The *Tool Enable* paramter of Patmax_OH_S is linked to the result output of Math_P_OH_S. If the above condition is fulfilled, then Patmax_OH_S starts to operate on the image to find the smaller OH-prefixes. The Tool Enable parameter of OCR_OH_S is linked to the Patmax_OH_S *Tool Enable* parameter and parameter storing the Pass result. The two paramters are combined by Math_OCR_OH_S as shown on Listing 2.

```
Patmax_OH_S.Tool_Enabled && Patmax_OH_S.Pass
```

Listing 2. The code for Math_OCR_OH_S

This expression ensures OCR_OH_S gets turned on if and only if Patmax_OH_S is turned on and succed to locate its trained pattern. If OCR_OH_S is turned on, it continues its segmentation and classifying operation as described for OCR_OH_L.

Patmax_OH_D_N starts operating on the image if one of the preceding tools fail. The preceding tools include Patmax_OH_L, Patmax_OH_S, OCR_OH_L and OCR_OH_S. Math_P_D_N combines the ouputs of these tools as shown on Listing 3.

```
Math_P_OH_S.Result && (Patmax_OH_S.Fail || OCR_OH_S.Fail ||
(OCR_OH_S.Result_Length<5) || (OCR_OH_S.Result_Length>6))
```

Listing 3. The code for Math_P_D_N

If PatMax_D_N fails, then the ARI could not be located by the pattern models trained for the three PatMax tools and whole process of the job ends with failure to read the ARI.

Finally, OCR_D_N operates on the image if PatMax_D_N succeed to locate the ARI. It is turned on by the output it receives form Math_OCR_D_N. The expression for Math_OCR_D_N is shown on Listing 4.

```
Pamax_D_N.Tool_Enabled && Pamax_D_N.Pass
```

Listing 4. The code for Math_OCR_D_N

5 Results and Discussion

5.1 Results from Preliminary Inspection of Images

After having the preliminary inspection and pixel-wise measurement, the following observations were made.

Foreground (text) Variations

- *Font style:* All characters in the same ARI have the same stroke size. There is no restriction to the type of font style used. Generally, the major font styles noted from the collected pictures are military, round, semi round and geometric-sans-serif. *Appendix 5* shows the font styles mentioned. There are also other font styles with a slight difference with aforementioned styles.
- *Size*: According to the pixel character height measurement for the ARIs, the range was 8 to 35 pixels.

- *Angle*: variation in rotation angle and skew angle were noticed. The range for characters' rotation is up to ± 30° as measured from horizontal axis. The skew angle range is up to ± 20° which is measured from vertical axis.
- *Polarity*: From the grey scale images of the aircraft, black text on white background (BW) and white text on back ground (WB) were observed (Appendix 1, Figure 23).
- *Prefix Variation*: most ARIs start with **OH-** (prefix assigned to Finland), some start with **D-** (Prefix assigned to German) and some ARIs start with **N** followed by number (prefix assigned to United States).

In addition to the above variations, the number of characters in a given ARI are either five or six. It is not less than five and greater than six.

Background Variations

There is a vast variation among the backgrounds of the ARIs (*Appendix 1 and Appendix 2*). The backgrounds are very noisy and consist of different features that might mislead, confuse or halt the OCR tool from segmenting and reading the characters properly. The sources of noise for background include

- *Color variations*: different aircrafts have different text background color. Besides, some aircrafts have two or more than two color painting which is the background for the ARI characters.
- *Illumination*: In some aircrafts uneven illumination is observed. This is mainly caused by the appearance of the aircrafts against the light source and their surface reflectance. Partially shadowed and extreme brightened ARIs are the results of uneven illumination.
- *Non-ARI texts*: There are also other characters on the body of the aircrafts (For instance, promotions, websites, company names and logos) which are not part of the ARI (*Appendix 2*).
- *Extraneous features and curvatures* that resemble characters and can be segmented as a text. Moreover, Lines through and around characters can create fragments that can merge with character fragments during grouping stage of the OCRMax and give the character a shape that cannot be recognized by matching with trained fonts.

5.2 Job Test Results

Patmax Performance

The Patmax performance is evaluated based on the criteria whether it is able to locate the pattern models it is trained to find or not. *Appendix 6, Table 6*: shows the performance result collected from a test made on In-sight explorer. A ‘Pass’ or letter ‘P’ is assigned for those aircraft images on which Patmax has located its model pattern. A ‘Fail’ or letter ‘F’ is assigned for those aircraft images that Patmax failed to find its trained model pattern. According to this criteria, the Patmax performance was 97.6%. *Appendix 7, Figure 26* shows images on which Patmax was able to find patterns. The higher score of the Patmax shows the 20 minimized model features chosen were represent the rest the 120 model features explained in *chapter 4*.

Factors that contributed for the 2.4% failure include background noises such as lines through and around texts, closer greyscale values of background and foreground. *Appendix 7, Figure 27* show aircraft images in which Patmax failed to locate the ARIs.

Accuracy of ROI Orientation

After the PatMax performance evaluation, the next evaluation was made on how accurately the ROI of the OCRmax was positioned based on the *fixture* (coordinate and angle) obtained from Patmax. For this evaluation two criteria are used; the coordinates and angle. The ROI location is compared to the ARI location (coordinate and angle). If ROI covers the whole ARI text area and its horizontal baseline (bottom line segment of the rectangle) is oriented approximately in the same angle with the base of the ARI, then ROI is evaluated us ‘Pass’, otherwise as ‘Fail’. Based on this criteria, the ROI orientation was successful on 93% of the 85 images tasted (*Appendix 6, Table 6)*. *Appendix 7, Figure 28 and Figure 29* show the successful ROI orientations and failed ROI orientations respectively.

ROI orientation depends on the *fixture* passed from Patmax to OCRmax. Consequently, the factors contributing for inaccurate ROI orientation result from incorrect coordinate and angle value obtained from Patmax.

OCRmax Performance

The performance of the OCRmax is evaluated based on accuracy of the string it returns compared to the characters of ARI it returns. According to the Pass and Fail evaluation test made based on this criteria, the score of OCRmax was 84.7% *(Appendix 6, Table 6)*. *Appendix 7, Figure 30 and Figure 31* shows aircraft images in which the ARI was read successfully and images where the developed job failed to read ARIs accurately.

In those image where ARIs were read successfully, OCRmax had managed to segment and classify characters accurately regardless of the foreground variations (such as, font style, size, polarity and angle) and background noises (such as color or grey scale variations, illumination defects, non-ARI texts, lines and extraneous features).

The major factors that contributed for the most failures of the OCRmax are the orientation and area coverage ROI. Apparently, an incorrect fixture obtained from Patmax results in incorrect ROI orientation which in turn causes inaccurate enclosure of the characters leading for the failure the OCRmax to segment and classify the text region accurately.

According to In-sight explorer manual, the ideal ROI text area coverage is configured in such a way that region should be extended by at least half the width of the widest character on the right and left, while extending at least a stroke width on the top and bottom [7]. Even though there were two OCRmax tools included in the *job* based on character sizes, the ideal area enclosure could not be kept for all ARIs as the ROI has fixed area coverage once it is defined in a given OCRmax tool. As a result, those text regions with small fonts face higher possibility of enclosing non-text fragments. The sources of the non-text fragments include lines through and around the texts, decorations and multiple-color backgrounds. In most images these fragments (noises) are ignored by *segmentation-parameters* and *advanced-parameters* defined based on the character properties. But some fragments still persist to contribute for failure of the job in reading the ARIs correctly.

In some images, the foreground (text) and background have close grey scale values, as a result both fall in the range either above or below the binarization threshold which at the end results the foreground and background being merged to one fragment. This apparently creates inaccurate segmentation or no segmentation at all as seen on some images (Appendix 7, Figure 31).

The main challenge in setting up parameters for both PatMax and OCRMax was tuning the parameters to the value that works for all ARIs, which is impossible for some parameters such as binerization threshold value, normalization. Rather, the values were tuned so that they work for most ARIs.

Note: The performance evaluation is not made with the intention of evaluating Cognex vision tool products. It shows only the performance of the *job* developed using the Cognex machine vision platform. The performance of the *job* depends on different factors, such as the experience of the developer, the problem being dealt, the software's algorithm in relation to the system requirement etc. As a result, the performance evaluation made here describes only the performance of the solution (*job*) developed for the AAR.

6 Conclusion

The objective of this thesis was to develop a feasible solution for an automated aircraft identification system based on Cognex machine vision platform. After classifying the system in to three subsystems, the scope was set to the vision subsystem in which aircraft identifiers (AIRs) are read with the application of OCR.

In-sight explorer was used in Emulator mode in which the selected vision camera was simulated offline without the presence of the vision camera. Aircraft images taken by normal cameras were collected and edited to develop a solution and carry out a test.

Patmax and OCRmax were the major cognex vision tools used to develop the required job. From the job test made on the aircraft images, in 97.6 % of the images the ARIs were located correctly and in 84.7% of the images ARI strings were read correctly. As a result, the overall performance of the developed *Job* is 84.7%. This result is promising to continue working on Cognex platform to develop a fully functional automated aircraft identification system that can read the ARIs with 100% accuracy.

This project can be continued further by improving the accuracy of the job developed using cognex platform and by designing the remaining two subsystems; the trigger and communication subsystems. Other machine vision platforms such as Matlab computer vision toolbox and Omron vision system can also considered as alternative platforms.

References

[1] Samantha F. Introduction to Machine Vision [Online]. Cognex Corporation. USA: 2015

URL: www.cognex.com/global/DownloadAsset.aspx?id=16341

Last accessed 20 January 2017

[2] Machine vision Introduction [Online]. SICK IVP. Version 2.2, December 2006

URL: https://www.sick.com/medias/Machine-Vision-Introduction2-2-web.pdf

Last accessed 2 February 2017

[3] Rafael C.Gonzalez and Richard E. Woods. Digital Image Processing [Online], Third Edition: USA: Pearson Prentice Hall; 2008.

[4] Line Eikvil. OCR: Optical Character Recognition [Online]. 1993

URL: https://www.nr.no/~eikvil/OCR.pdf

Last accessed 4 February 2017

[5] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Sureshm Tao Wang, David J. Wu, Andrew Y. Bg. Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning [Online]. Stanford University. USA

URL: https://crypto.stanford.edu/~dwu4/papers/ICDAR2011.pdf

Last accessed 4 March 2017

[6] Julinda Gllavata, Ralph Ewerth and Bernd Freisleben. A Robust Algorithm for Text Detection in Images [Online] University of Siegen, Germany

URL: http://www.mathematik.uni-marburg.de/~ewerth/papers/ISPA2003.pdf

Last accessed 4 March 2017

[7] Teofilo E. de Campos, Bodla Rakesh Babu, Manik Varma. Character Recognition in Natural Images [Online]

URL: http://personal.ee.surrey.ac.uk/Personal/T.Decampos/papers/decampos_etal_visapp2009.pdf

Last accessed 4 March 2017

[8] Finnish Transport Safety Agency: Aircraft Registration

URL: http://www.trafi.fi/en/aviation/aircraft_register/aircraft_registration_marks

Last accessed 20 January 2017

[9] Cognex Corporation. In-Sight Installation and Operation: Manual V5.2: Introduction to hardware [Online]; Cognex Corporation. USA; September 2016

URL: http://www.cognex.com/support/downloads/File.aspx?d=3324

Last accessed 13 March 2017

[10] In-Sight Explorer [Computer Program] V5.3.0 (722). USA: Cognex Corporation; 2016.

[11]Aircraft photos Collected from the following websites

URL: https://www.jetphotos.com/

URL: http://www.airliners.net/

Last accessed 16 December 2016

Variation in location of ARI (ROI), font styles, size, polarity, rotation, skew

Figure 22:Aircraft images showing foreground and background variations

Figure 23: Polarity Variation on grey scale; White text on black bacground-WB (left) and Black text on White background-BW (Right)

Texts that are not part of the ARI

Figures below show nonARI texts on the aircrafts

Figure 24: Aircraft images showing nonARI texts

Parameter Definitions

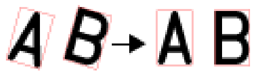
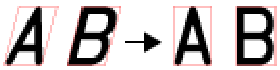
The following PatMax and OCRMax parameter definitions are taken from In-Sight explorer software help manual. [10]

Table 2: Patmax Parameter Definitions

Parameter name	Description
Tool Image	Defines which image the tool will utilize to perform its inspection; the unfiltered, acquired image (the default setting), or the output image of an Image Filter Tool.
Tool Fixture	Defines a fixture for the tool. This control is only enabled if another tool that defines a fixture has already been added.
Mode	Defines the operational mode of the tool: *Identify* or *Verify*.
Accept Threshold	Defines the degree of similarity that must exist between the *Model* pattern and the found pattern (0-100; default = 65).
Contrast Threshold	Defines the minimum acceptable contrast that must be present in the found pattern(valid parameter range is 0-100)
Angle Tolerance	Defines how far the found pattern can be rotated from the position of the Model pattern and still be recognized as a valid pattern (± 0-180°)
Scale Tolerance	Defines the allowable percentage of scaling (size variations) between the found pattern and the Model pattern (0-50;)
Difference Accept	Defines the allowable difference in scores that can exist between the found pattern and any of the trained Model patterns (0-20). The value is the score difference between any two trained *Model* patterns
Strict Scoring	Defines whether or not missing or occluded features of the found pattern will affect the score
Timeout	Defines the amount of time, in milliseconds (0 to 30,000), that the tool will search for pattern(s) before execution is stopped and the tool returns a Fail.

Table 3: OCRMax Parameter Definations

Parameter name	**Description**
General Tab	
Tool Image	Defines which image the tool will utilize to perform its inspection; the unfiltered, acquired image (the default setting), or the output image of an Image Filter Tool.
Tool Fixture	Defines a fixture for the tool. This control is only enabled if another tool that defines a fixture has already been added
Tool Enabled	Defines when and whether or not the inspection tool should run
Setting Tab	
Font Library	Defines a font database reference for the tool.
Inspection Mode	Defines the operational mode of the tool: *Read* or *Read/Verify*.
Accept Threshold	Defines the minimum acceptable match score for each character. Any character with a match score below the *Accept Threshold* will fail.
Confusion Threshold	Defines the minimum difference required between the match scores of the highest scoring character and the second highest scoring character
Segmentation tab	
Character Polarity	Defines the polarity of the characters in the input image: *Black on White*, *White on Black* or *Auto.*
Character Width Type	Defines how the widths of the characters in the font are expected to vary: *Auto (default)*, *Fixed* or *Variable*.
Minimum Character Width	Defines the minimum width of a character's character rectangle, in pixels, that a character must have to be reported.
Minimum Character Height	Defines the minimum height of a character's character rectangle, in pixels, that a character must have to be reported.
Use Maximum Character Width	Defines whether or not the tool should account for the maximum allowable width of a character's character rectangle. When enabled, a character wider than the specified value will be split into segments that are not too wide.
Maximum Character Width	Defines the maximum width of a character's character rectangle, in pixels, that a character must have to be reported.
Use Maximum Character Height	Defines whether or not the tool should account for the maximum allowable height of a character's character rectangle
Use Minimum Character Aspect Ratio	Defines whether or not the function will account for the minimum allowable aspect ratio of a character, where the aspect ratio is defined as the height of the entire line of characters, divided by the width of the character's character rectangle

Angle Range	Defines the rotation angel search range (0 - 45), in degrees. 
Skew Range	Defines the skew search range (0 – 45) in degrees. 
Advanced Tab	
Character Fragment Merge Mode	Defines how the tool should merge character fragments when forming characters during segmentation: Require Overlap, Set Min Inter-Character Gap or Set Min Inter-Character Gap/Max Intra-Character Gap.
Minimum Character Fragment Overlap	Defines the minimum fraction by which two character fragments must overlap each other in the X direction, in order for the two fragments to be considered as part of the same character.
Max Intra-Character Gap	Defines the minimum fraction (0 - 100) by which two character fragments must overlap each other in the X direction, in order for the two fragments to be considered as part of the same character.
Max Intra-Character Gap	Defines the maximum gap size, in pixels (0 - 1000), that can occur within a single character, even for damaged characters.
Min Inter-Character Gap	Defines the minimum gap size, in pixels that can occur between two characters.
Minimum Character Fragment Size	Defines the minimum number of foreground that a fragment must have in order to be considered for possible inclusion in a character. A character fragment is a blob in the binarized image.
Normalization Mode	Defines the mode used to normalize the image: *None*, *Global*, *Local* or *Local Advanced.*
Use Stroke Width Filter	Defines whether or not to remove from a normalized image everything that does not have the same stroke width as the as the rest of the image
Ignore Border Fragments	Defines whether or not the function will completely ignore any fragments that touch any border of the region
Binarization Threshold	Defines a percentage modifier (0 - 100; default = 50) in the range that is used to compute the binarization threshold, in the normalized image, that binarizes the image between foreground and background.
Character Fragment Contrast Threshold	Defines the minimum amount of contrast [in normalized image grayscale levels that a fragment must have, relative to the Binarization Threshold, in order to be considered for possible inclusion in a character
Maximum Fragment Distance To Mainline	Defines the distance, in pixels that a fragment may be removed from the "mainline" running horizontally through the text.
Segmentation Analysis Mode	Defines the type of character analysis mode to perform to determine the optimal character segmentation: Minimal or Standard

Minimum Pitch	Defines the minimum pitch, in pixels (0 - 1000; default = 0), that can occur between two characters, where the pitch is computed based on the *Character Pitch Position* parameter.
Character Pitch Position	Defines how the pitch between characters will be measured: *Auto (default)*, Left-To-Left, Center-To-Center or Right-To-Right.
Character Pitch Type	Defines the metric used to define the spacing of characters: *Auto (default)*, *Fixed*, *Proportional* or *Variable*.
Spaces Tab	
Find Spaces	Defines how the tool will handle the insertion of space characters into gaps between other characters: None (default), Insert Single Space or Insert Multiple Spaces.
Space Score Mode	Defines how the function will calculate scores for spaces: Always Score 100 (default) or Score Based on Clutter.
Space Minimum Width	Defines the minimum width of a space character, in pixels (0 - 1000; default = 10).
Space Maximum Width	Defines the maximum width of a space character, in pixels (0 - 1000; default = 100).
Variable Length Tab	
String Length	Defines whether or not fielding will be run in fixed-length (default) or variable-length mode.
Maximum String Length	Defines the maximum acceptable string length
Minimum String Length	Defines the minimum acceptable string length
Maximum Last Fielded Index	Defines the fielding subsequences to be considered, which must end at a position that is no smaller than this index value

Parameter Configuration

Table 4: Patmax parameter configuration

Parameter name	Patmax __OH_L	Patmax __OH_S	Patmax_ D_N	Description
	General Tab			
Tool Image	Acquisition. image	Acquisition. image	Acquisition. image	The original image acquired from a camera, which is not processed by another vision tool, is used.
Tool Fixture	None	None	None	It does not need a fixture as input, as it is supposed to find a fixture for its trained model.
Tool Enabled	ON	Expression	Expression	Every time the program starts, the first tool that operates on the images is Patmax_OH_L. As a result, Patmax_OH_L is always on. The rest of tools are enabled depending on the results of Patmax and OCRmax tools. *Expression* in this case is Boolean argument passed from one or more than one tools in order to enable or disable the receiving tools.
	Setting Tab			
Mode	Identify	Identify	Identify	The tool is needed to identify the model and report the fixture of the model found, not just to verify that the model exists in the image.
Accept Threshold	70	70	70	It is tuned to the smallest value that avoids wrong matching. It is set with repetitive testing.
Contrast Threshold	0	0	0	Due to illumination variation there is no threshold value which could be taken with certainty to represent the minimum contrast threshold.
Angle Tolerance	20	20	20	Range is set based on preliminary image inspection results.
Scale Tolerance	23%	48%	30%	Models of Patmax_OH_L have height of 30 pixels and Models of Patmax_OH_S have height of 16 pixels. The scale tolerances are arranged in such a way that Patmax_OH_L covers ARIs having character height range 23 – 37 (30±23%) and Patmax_OH_S covers ARIs having character height range 8 – 23 (16±48%) approximately. Using this configuration, the whole character size range by height (8 – 35) is covered. For Patamx_D_N 30% or even less scale can cover the whole range.

Strict Scoring	Un-checked	Un-checked	Un-checked	Disabling *Strict Scoring* option allow the tools to approximate deformed patterns with trained patterns.
Timeout (mil-lisconds)	30000	30000	30000	30 second is the maximum execution time allowed on Patmax. The maximum execution time is set as the time between two flights (two aircraft identifications) can accommodate more than 30 second execution time.

Table 5: OCRmax Parameter Configuration

Parameter name	**OCR_ OH_L**	**OCR_ OH_S**	**OCR_ D_N**	**Description**
	General Tab			
Tool Image	Acquisition. image	Acquisition. image	Acquisition. image	The original image acquired from a camera is used.
Tool Fixture	Patmax _OH_L. Fixture	Patmax _OH_S. Fixture	Patmax _D_N.Fixture	Each OCRmax tools accept a fixture (angle and coordinates of the ROI) from their corresponding Patmax tool.
Tool Enabled	Expression	Expression	Expression	The tools run based on the pass/Fail result of their corresponding Patmax tools and the conditional statements from the Math tool.
	Setting Tab			
Font Library	Trained Font	Trained Font	Trained Font	Trained font Library created for this project is used.
Inspection Mode	Read	Read	Read	OCRmax in this application is needed for character recognition, not character verification
Accept Threshold	60%	60%	60%	Minimum acceptable match score avoiding wrong character read for each character is set with test done repeatedly.
Confusion Threshold	0	0	0	The highest scoring character is accepted regardless of the score of the second highest scoring character.
	Segmentation tab			
Character Polarity	Auto	Auto	Auto	Input images have both polarities: *Black on White*, *White on Black*. As a result the *Character Polarity* set to *Auto* so that the tools decides the type of the polarity
Character Width Type	Variable	Variable	Variable	Character sizes vary and so does the *Character Width*
Minimum Character Width	4	3	4	The character resulting minimum character rectangle width is **'I'.** The values are set by pixel counting.
Minimum Character Height	20	7	20	The minimum height of a character's rectangle as measured by pixel counting.

Use Maximum Character Width	Checked	Checked	Checked	This option is checked so that the OCRmax tools account for the maximum allowable width of a character's rectangle.
Maximum Character Width	35	25	35	The widest characters are **'M'** and '**W'.** The values for each OCRmax tool is set by pixel counting.
Use Maximum Character Height	Checked	Checked	Checked	This option is checked so that the OCRmax tools account for the maximum allowable height of a character's rectangle.
Maximum Character Height	37	23	37	Measured by pixel counting.
Angle Range	20°	20°	20°	The largest rotation angle measured from horizontal axis was 18.36°. So, the range was set to include this maximum value.
Skew Range	30°	30°	30°	The largest skew measured from vertical axis was 29.36°. So, the range was set to include this maximum value.
	Advanced Tab			
Character Fragment Merge Mode	Set Min Inter-Character Gap/Max Intra-Character Gap.	Set Min Inter-Character Gap/Max Intra-Character Gap.	Set Min Inter-Character Gap/Max Intra-Character Gap.	By selecting this option the tool will account for both minimum inter-character gap and maximum Intra-Character Gap (especially for characters separated into fragments during the image processing.
Minimum Character Fragment Overlap	3	3	3	This value is assigned for the OCRmax tools to account for kerning (Character region overlap).
Max Intra-Character Gap	2	2	2	Even though ARI character strokes are interconnected, during binerization strokes of one character may get separated in to two fragments. The max intra-character gap will account for the space that could appear between the fragments of the same characters. The gap is measured by pixel counting and the maximum result found was 2.
Min Inter-Character Gap	1	1	1	The minimum space between characters was found to be 1 as measured by horizontal pixel counting.

Minimum Character Size	60	21	60	The minimum character size can be found by multiplying the minimum character height and the minimum character width, which are already mentioned above.
Normalization Mode	Local Advanced	Local Advanced	Local Advanced	Local Advanced was chosen because in addition to using information about each local character region in the ROI to normalize the image, it also adjusts for inconsistent text contrast.
Use Stroke Width Filter	Checked	Checked	Checked	In a given ARI strokes are uniform. By selecting the Use Stroke Width Filter, everything that does not have the same stroke width as the rest of the image is removed from a normalized image.
Ignore Border Fragments	Checked	Checked	Checked	The ROI dimensions are set to accommodate the largest ARIs. Consequently, if any fragment appears on the border then it has be extraneous feature. By selecting the Ignore Border Fragments option, the OCRmax tools will completely ignore any fragments that has contact with the border of the region
Binarization Threshold	50	50	50	Since the Character Polarity is set Auto, the binerization threshold is equally divided.
Character Fragment Contrast Threshold	30	30	30	The grey scale picture obtained from the original RGB aircraft images have very wide contrast variation. With repeated tests, the value is set to be 30
Maximum Fragment Distance To Mainline	5	3	5	All ARI characters are upper-case letters. As a result, in a given ARI the characters are expected to have the same distance from the main line. These values are assigned to account for possible vertical drifts that could occur when characters are printed on the aircraft body
Segmentation Analysis Mode	Standard	Standard	Standard	Among the two choices (*Standard* and *Minimal*), standard choice is chosen as it is more advanced and does optimal segmentation using additional parameters such as character spacing and pitch.
Minimum Pitch	5	4	5	The minimum pitch measured from left to left can be calculated by adding *Minimum Character Width* and *Minimum Inter-Character Gap*

Character Pitch Position	Left-To-Left	Left-To-Left	Left-To-Left	Left-To-Left pitch measurement was more convenient.
Character Pitch Type	Variable	Variable	Variable	Due to the font variation, the pitch type is also variable.
	Spaces Tab			
Find Spaces	None	None	None	Space parameters are not used as there is no word-space to be in considered in ARIs. For the remaining space parameters the default values are not changed.
Space Score Mode	default value	default value	default value	Not Used
Space Minimum Width	default value	default value	default value	Not Used
Space Maximum Width	default value	default value	default value	Not used
	Variable Length Tab			
String Length	Variable	Variable	Variable	The string length varies between 5 and 6 excluding the hyphen.
Maximum String Length	6	6	6	
Minimum String Length	5	5	5	
Maximum Last Fielded Index	default value	default value	default value	not used

Font Templates for OCRmax Font Training

Round

Semi-round

Military

Geometric-sans-sarif

Figure 25: Font Templates

Results from the Job Performance Test

P=Pass F=Fail

Table 6: Results from the job performance test

No	Aircraft image names by ARI	Patmax (Location)	ROI Position	OCRmax (OCR)
1	OH-ACM	P	P	P
2	N653TX	P	P	P
3	OH-OBO	p	P	P
4	OH-XIU	p	P	P
5	N5519Z	p	P	P
6	OH-DIR	p	P	P
7	OH-DDS	P	P	P
8	OH-PYM	P	P	F
9	OH-COF	p	P	P
10	OH-GSA	P	P	P
11	OH-OUI	P	P	P
12	D-ECTL	P	P	P
13	D-EJHE	P	P	P
14	OH-U221	F	F	F
15	OH-CIO	P	P	P
16	OH-SUN	P	P	P
17	OH-BCK	p	P	P
18	OH-EPA	p	P	P
19	OH-U520	p	P	P
20	OH-HVH	p	P	P
21	OH-U645	p	P	P
22	OH-AYY	p	P	P
23	OH-CHZ	P	P	P
24	OH-PMK	p	P	P
25	OH-CIQ	p	P	P
26	OH-KAS	p	P	P
27	OH-CGD	p	P	P
28	OH-CIO	p	P	P
29	OH-MIV	p	P	F

30	OH-PJJ	p	p	P
31	OH-CMY	p	P	F
32	OH-TJS	p	P	P
33	OH-AYO	p	P	F
34	OH-BDX	P	P	P
35	OH-CDI	P	P	P
36	OH-CTE	p	P	P
37	OH-GSB	P	P	P
38	OH-CAY	P	P	F
39	OH-CGD	P	P	P
40	OH-CIO	P	F	P
41	OH-CSG	P	P	P
42	OH-CVP	P	P	P
43	OH-DBS	P	P	P
44	OH-G016	P	P	F
45	OH-KAM	P	P	P
46	OH-MXN	p	F	P
47	OH-PIJ	p	P	P
48	OH-PNX	P	P	P
49	OH-SIA	P	P	P
50	OH-STL	P	F	F
51	OH-TAE	P	P	P
52	OH-TIL	P	P	P
53	OH-U345	P	P	F
54	OH-U362	P	P	P
55	OH-U466	P	P	P
56	OH-U486	P	P	P
57	OH-U544	p	P	P
58	OH-U577	F	F	F
59	OH-U600	P	F	F
60	OH-U620	P	P	P

61	OH-U645	P	P	P
62	OH-TIL_2	P	P	P
63	OH-VFR	P	P	P
64	OH-VTP	P	P	P
65	OH-XRT	P	P	P
66	OH-XSR	P	P	P
67	OH-XTH	P	P	p
68	OH-XXL	P	P	F
69	OH-XXV	P	P	P
70	OH-PYW	P	P	P
71	OH-KAS	P	P	P
72	OH-PVD	P	P	F
73	OH-HVM	P	P	P
74	OH-CCY	P	P	P
75	OH-KAU	P	P	P
76	OH-AWB	P	P	P
77	OH-DBS(N)	P	P	P
78	OH-PAX	P	P	P
79	OH-PTC	P	P	P
80	OH-SRH	P	P	P
81	OH-U530	P	P	P
82	OH-DAC	P	P	P
83	OH-CIP	P	P	P
84	OH-CRH	P	P	P
85	OH-MOI	P	P	P
	Performance Evaluation	97.6%	93%	84.7 %

Pictures Showing Results from the Job Performance Test

Examples of aircraft images in which Patmax located ARI succussfuly are shown below.

Figure 26: Examples of aircraft images in which Patmax located ARI succussfuly

Images of aircrafts where Patmax has failed to locate ARIs are shown below

Figure 27: Images of aircrafts where Patmax has failed to locate ARIs

Figures showing ROI Pass and Fail results

Figure 28: Successful ROI Orientation

Figure 29: Failed ROI Orientation

Figures showing OCRmax Pass and Fail Results

Figure 30: Images showing successful OCRmax reading

Aircraft images showing incorrect ARI reading

Figure 31: Aircraft images showing failed or incorrect ARI reading