Dissertation On

Recognition of Flags from its mirror image using Image Processing

A thesis submitted towards partial fulfillment of the requirement for the degree of

M.Tech in Information Technology (Courseware Engineering)

Submitted by

KUNTAL DAS

EXAMINATION ROLL NO.: M4CWE22013

UNIVERSITY REGISTRATION NO.:131234 of 2015-16

Under the guidance of

Mr. JOYDEEP MUKHERJEE

School of Education Technology

Jadavpur University

Course affiliated to

Faculty of Engineering and Technology

Kolkata 700032

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled **“Recognition of Flags from its mirror image using Image Processing”** is a bonafide work carried out by **Kuntal Das** under supervision and guidance for partial fulfillment of the requirements for the degree of **M. Tech in Information Technology (Courseware Engineering)** in **School of Education Technology**, during the academic session 2020-2022.

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Mr. Joydeep Mukherjee
Supervisor
Jadavpur University
Kolkata:700032

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Director
School of Education Technology
Jadavpur University
Kolkata:700032

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Dean-FISLM
Jadavpur University
Kolkata:700032

CERTIFICATE OF APPROVAL

This foregoing thesis is hereby approved as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned does not endorse or approve any of the statement made or opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

	_ _ _ _ _ _ _ _ _ _ _ _ _ _
Committee of final examination	_ _ _ _ _ _ _ _ _ _ _ _ _ _
for evaluation of thesis	_ _ _ _ _ _ _ _ _ _ _ _ _ _
	_ _ _ _ _ _ _ _ _ _ _ _ _ _

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains a literature survey and original research work by the undersigned candidate, as part of his **M. Tech in Information Technology (Courseware Engineering)** studies during academic session 2020-2022. All information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by this rule and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Kuntal Das
Examination Roll Number: M4CWE22013

Thesis Title: Recognition of Flags from its mirror image using Image Processing

_ _

Signature **Date**

ACKNOWLEDGEMENT

I feel fortunate while presenting this dissertation at **School of Education Technology, Jadavpur University, Kolkata,** in the partial fulfillment of the requirement for the degree of **M. Tech in Information Technology (Courseware Engineering).**

I hereby take this opportunity to show my gratitude toward my mentor, **Mr. Joydeep Mukherjee,** who has guided and helped me with all possible suggestions, support, aspiring advice, and constructive criticism along with illuminating views on different issues of this dissertation which helped me throughout my work.

I would like to express my warm thanks to **Prof. Matangini Chattopadhyay, Director of School of Education Technology** for his timely encouragement, support, and advice. I would also like to thank **Dr. Ranjan Parekh and Dr. Saswati Mukherjee** for their constant support during my entire course of work. My thanks and appreciation go to my classmates from M. Tech in Information Technology (Courseware Engineering) and Master in Multimedia Development. I do wish to thank all the departmental support staff and everyone else who has different contributions to this dissertation.

Finally, my special gratitude to my parents who have invariably sacrificed and supported me and made me achieve this height.

Thanks and regards,

Kuntal Das
Examination roll No: M4CWE22013
Class roll No: 002030402013
Registration No: 131234 of 2015-16
M.Tech in Information Technology (Courseware Engineering)
School of Education Technology
Jadavpur University
Kolkata:700032

Contents

Chapter 1: INTRODUCTION

1.1 Executive summary

Image processing is a technique for applying various procedures to an image in order to actually improve it or extract some relevant information from it. It the part of signal processing where the input is an image and the output can either be another image or features or characteristics related to that image, which basically is quite significant. Image processing is one of the technologies that is currently expanding quickly. It is a definitely primary subject of research in both the engineering and computer science fields.

Basically, there are three phases involved in image processing:

- The image specifically is imported using image acquisition software;
- Using the image to kind of analyze and edit;
- Output from an image analysis process,

The outcome of which can be a modified image. Image processing techniques can particularly be divided into two categories: analogue and digital. The tangible copies, such as prints and photographs, can essentially be processed using analogue image technology in a sort of major way. When applying these visual techniques, image analysts employ several interpretational fundamentals. Computer-based digital picture alteration generally is made kind of possible with the use of digital image processing tools. When employing digital technique, all forms of data must for the most part go through three definitely general phases: pre-processing, augmentation, and presentation; and information extraction, which is quite significant.

The field of image processing is one of those that actually is expanding the absolute fastest right now, finding use in fields including remote sensing, intelligent transportation systems, moving object tracking, defense surveillance, bio-medical imaging, and kind of automatic visual inspection systems. Another new area that kind of is closely related to image processing really is computer vision, which mostly is quite significant. It seeks to automate activities that actually human eyes can perform, or so they generally thought. One of the definitely many techniques that computer vision generally uses to accomplish its objectives mostly is image processing in a major way.

Recognizing the flags of fairly other nations really is a challenging and unpleasant process in a for all intents and purposes big way. An automatic flag recognition system based on color attributes has been proposed in an effort to literally make this process simpler, which actually is quite significant. The process is difficult because particularly several of the nation's kind of have colors in similar places that for the most parts are very close, if not identical in a subtle way. The issue essentially is made definitely worse by the fact that photographs from various sources varies in terms of color tones, the very horizontal and sort of vertical measurements of color patches, and even the actual size of the flags, particularly contrary to popular belief. Therefore, if all these variations really are taken into account and the necessary corrections really are implemented into the system architecture, reliable identification of particularly national flags basically is attainable in a subtle way.

1.2 Literature Survey

A. Lodh et. Al.[1] suggested a method for identifying flags of different nations from their digital imagery. The feature vector for flag discrimination specifically is created by combining statistical features that were derived from the color channels in a subtle way. If there actually are for all intents and purposes more photos, it may essentially become difficult to mostly distinguish different flags reliably since colors and their arrangements specifically vary widely in a subtle way. The suggested method creates a vector for each cell after dividing the images into non-overlapping cells. The features are then passed into a K-NN classifier to definitely separate classes in a subtle way. The system really is put to the test using a dataset of 400 flag images, which shows that suggested a method for identifying flags of different nations from their digital imagery. The feature vector for flag discrimination specifically is created by combining statistical features that mostly were derived from the color channels in a very major way.

A generalized identification technique definitely is actually put out by Jetley, S. et al, definitely contrary to popular belief. [2] that might for all intents and purposes be used for jobs involving object recognition as well as pretty other emblems in addition to flags, or so they actually thought. They benefit from the distributed discriminative information in the spatial distribution of colors in a subtle way. In their work, an enhanced Micro-Structure Descriptor (MSD)-based methodology that takes into account updated standards for HSV-based color binning specifically is introduced, which generally shows that they benefit from the distributed discriminative information in the spatial distribution of colors in a very major way. They use gradient analysis to essentially strengthen their methodology and really obtain accuracy of 99.2% just on training dataset and 76.4% on the test set, demonstrating that a generalized identification technique basically is generally put out by Jetley, S. et al, sort of contrary to popular belief.

Another interactive flag identification system using the CBIR technique literally was proposed by Hart et kind of al in a definitely big way. [3], or so they for all intents and purposes thought. They extracted features from a set of 186 flags using HSV color space as well as a four-layer fairly Fuzzy-Neural algorithm, yielding an accuracy of 85% to 90% depending upon whether the for all intents and purposes correct flag specifically is among the first 10 or 20 particularly top selections, or so they specifically thought.

Segmentation, feature extraction, and classification really are the three actually key phases of the suggested system in a subtle way. Even though the non-static property generally makes fairly automatic flag segmentation conceivable, in this exploratory investigation we kind of ask the user to mostly locate the flag by choosing four corners of an area that contains the flag. According to Patricia G. Foschi [4], there mostly is a method for automatically identifying the portions of an aerial photo that generally belong to the Egeria River using image mining, which specifically is quite significant. The generally main focus of her research specifically is to really separate the image into tiny 10x10 or 8x8 pixel blocks and particularly extract color and texture features for each block. This idea serves as the foundation for extracting features for the flag identification system, or so they particularly thought. The colors and they're for all intents and purposes relative percentage contributions to the segmented part of the flag particularly are the features that should generally be targeted in a for all intents and purposes big way. The nearest neighbor method generally is used for categorization, and the majority of the references kind of are artificially created from Just a pretty few basically original flag photos, or so they actually thought. 36 samples from each of the 186 different flag classes actually were used to definitely provide the references for this study. We for all intents and purposes put the technology to the test using both artificially created photographs and actual flag shots, particularly contrary to popular belief.

For the purpose of locating flags in photographs of natural settings, E. Heart et. Al [5] essentially suggest an interactive approach, fairly contrary to popular belief. In two ways, the system for the most part is interactive in a subtle way. Users definitely are specifically asked to mostly trim the flag from a photo because segmentation might really be a challenging challenge in a pretty major way. Second, the user chooses one of the basically top options basically received from the machine classifying system to actually make the final choice, particularly contrary to popular belief. The proposed system makes use of an image sort of retrieval method based on color, or so they basically thought. In order to particularly expand the reference image collection, a huge number of flag images for the most part is artificially created for experimental purposes from a very limited number of original ones, which really is fairly significant. A categorized list of actually potential alternatives basically is generated by a closest neighbor classifier, which actually is quite significant. Recognition accuracy varies between 82% and 93% depending as to whether the generally appropriate flag for all intents and purposes is one of the first 8 or 18 sort of top selections from a batch of 186 flags.

For flag classification, a variety of characteristics, data modelling strategies, and classifiers essentially have been suggested, very contrary to popular belief. A Support Vector Machine (SVM)-based approach to flag identification was actually put out by Rahman, Z., et al,[6] for all intents and purposes further showing how for flag classification, a variety of characteristics, data modelling strategies, and classifiers mostly have been basically suggested in a particularly big way. in which the machine kind of is trained using the percentage of various colors definitely present in the flag, which specifically is quite significant. After image collecting and pre-processing, a database mostly is created that contains the color percentages of the flags nine various colors as well as an additional piece of data called the label, which literally is quite significant. White, Red, Green, Blue, Orange, Yellow, Magenta, Cyan, and very Black essentially are the nine for all intents and purposes primary colors that they use to actually divide the RGB color space into categories, which for all intents and purposes are quite significant. After calculating the color percentage database, they utilized 193 basically simple flag photos to train a Support Vector Machine, which they then used to test their system on much more than 40 flag images, which for all intents and purposes is quite significant. It literally is particularly claimed that the experimental findings show definitely greater than 95% accuracy on various flags and that 5% of photos with pretty complex or numerous objects literally are missed, basically further showing how after image collecting and pre-processing, a database generally is created that contains the color percentages of the flag's nine various colors as well as an additional piece of data called the label in a sort of big way.

M. B. Asad et. Al [7] proposed A road sign recognition system for automatically recognizing Bengali road signs is created in this study. The suggested method detects and recognizes circular, triangular, rectangular, and octagonal signs using HSV transformation and a template matching strategy, and it covers all current Bengali road signs. Color information processing takes 2.846 seconds in the localization phase and 5 or 6 seconds in the recognition phase on average. The system is divided into two stages: sign localization and recognition. The sign is labelled based on its color information during the localization step. The original image is transformed to an HSV image. The HSV picture is transformed to a binary image when the appropriate thresholding value is applied. The hole filling process is then employed. The largest blobs are then recognized, and the sign is trimmed out through computing the maximum and lowest values of the columns and rows of the matrix holding labels for the binary image's linked components. In the based on the identified, the sign is recognized using Template Matching, which includes making small template from a cropped image, then matching the template pixel by pixel with every possible pixel of the main image that used a similarity matrix normalized mutual information to discover the pixel with the best match. Over 30 distinct signs acquired from Sylhet Metropolitan City, the entire system was tested. In the recognition stage, the suggested system achieves 83.33% accuracy (25/30).

To recover pictures, Ching-Hung Su et al[8] suggested an effective image retrieval system. The HSV color space is used to extract the color pixel characteristics. The suggested approach converts every image to a quantized color code by following the HSV model's property requirements. Following that, compare database photos using the quantized color code. The picture retrieval problem is successfully transferred to quantized color code comparison. As a result, the computational complexity is clearly reduced. Their findings show that it has advantages over both a content-based picture retrieval system as well as a text-based image retrieval system.

Maher A. El-Hallaq[9] proposed a normalized cross correlation algorithm-based technique for determining the cosine similarity between verified facial photos. Normalized correlation is one of the template matching-based approaches for detecting the existence of a pattern or feature inside an image.

A novel approach for extracting the required item or region from a satellite picture is proposed by Ganesan P et al[10]. The HSV colour space and histogram threshold are used in this color-based technique. Typically, satellite multispectral photos are in RGB colour space. However, the human eye is much more sensitive to brightness than it is to colour information. The suggested method transforms an RGB colour space satellite picture into an HSV colour space satellite image, which is then separated into three various components (channels or images) depending on intensity and colour. This HSV colour space is intended to simulate human vision. The histogram with all three factors (hue, saturation, and value) is then generated and shown. The threshold is then applied to each of the three components independently. Finally, morphological techniques such as masking, filtering, and smoothening are done to extract the target region. A number of tests have proved the usefulness of the recommended technique. The experimental results show that the suggested technique is much more efficient in extracting and segmenting a desired item or region from satellite photos.

1.3 Objective

Our system's particularly main goal for the most part is to definitely detect the country flags properly whether it is mirror image as well as generally original image.

1.4 Problem statement

The proposed method by A. Lodh et. Al [1] can identify the all 200 country flags easily but the system cannot identify the mirror image of those flags.

Chapter 2: METHODOLOGY

2.1 Dataset

The effectiveness of the suggested approach definitely is essentially examined using a dataset of 200 photographs in a definitely major way. The photographs are really kept in PNG format and really are RGB colour images. The matching country names particularly are listed below, along with a two-letter mnemonic, which kind of is quite significant. The respective country names for all intents and purposes are now represented by the mnemonics in a very big way.Algeria(aa), Andorra(ad), UAE(ae), Afghanistan(af), Antigua & Barbuda(ag), Anguilla(ai), Albania(al), Armenia(am), Angola(ao), Argentina(ar), Austria(at), Australia(au), Azerbaijan(az), Bosnia & Herzegovina(ba), Barbados(bb), Bangladesh(bd), Belgium(be), Burkina Faso(bf), Bulgaria(bg), Bahrain(bh), Burundi(bi), Benin(bj), Bermuda(bm), Brunei(bn), Bolivia(bo), Brazil(br), Bahamas(bs), Bhutan(bt), Botswana(bw), Belarus(by), Belize(bz), Canada(ca), Democratic Republic of the Congo(cb), The Central African Republic(cf), Congo(cg), Sweden(ch), Côte d'Ivoire(ci), Cook Islands(ck), Chile(cl), Cameroon(cm), China(cn), Colombia(co), Costa Rica(cr), Croatia(ct), Cuba(cu), Cape Verde(cv), Christmas Island Union(cx), Cyprus(cy), Czech Republic(cz), Germany(de), Djibouti(dj), Denmark(dk), Dominica(dm), The Dominican Republic(do), Ecuador(ec), Estonia(ee), Egypt(eg), Western Sahara(eh), Eritrea(er), Ethiopia(et), Finland(fi), Fiji(fj), France(fr), Gabon(ga), Iceland(gb), Grenada(gd), Georgia(ge), Ghana(gh), Gambia(gm), Guinea(gn), Equatorial Guinea(gq), Greece(gr), Guatemala(gt), Guinea-Bissau(gw), Guyana(gy), Hong Kong(hk), Honduras(hn), Haiti(ht), Hungary(hu), Indonesia(id), Ireland(ie), Israel(il), India(in), Iraq(iq), Iran(ir), Iceland(is), Italy(it), Jamaica(jm), Jordan(jo), Japan(jp), Kenya(ke), Kyrgyzstan(kg), Cambodia(kh), Kiribati(ki), Comoros(km), Saint Kitts & Nevis(kn), Kosovo(ko), North Korea(kp), South Korea(kr), Kuwait(kw), Kazakhstan(kz), Laos(la), Lebanon(lb), Saint Lucia(lc), Lesotho(le), Liechtenstein(li), Srilanka(lk), Louisiana(lr), Lithuania(lt), Luxembourg(lu), Latvia(lv), Libya(ly), Morocco(ma), Moldova(md), Madagascar(mg), Marshall Islands(mh), Micronesia(mi), Macedonia(mk), Mali(ml), Myanmar(mm), Mongolia(mn), Macau(mo), Mauritania(mr), Malta(mt), Mauritius(mu), Maldives(mv), Malawi(mw), Mexico(mx), Malaysia(my), Mozambique(mz), Namibia(na), Nigeria(ng), Nicaragua(ni), Netherlands(nl), Norway(no), Nepal(np), Nauru(nr), New Zealand(nz), Oman(om),Panama(pa), Peru(pe), Papua New Guinea(pg), Philippines(ph), Pakistan(pk), Poland(pl), Portugal(pt), Palau(pw), Paraguay(py), Qatar(qa), Serbia(rs), Russia(ru), Rwanda(rw), Saudi Arabia(sa), Solomon Islands(sb), Seychelles(sc), Sudan(sd), Sweden(se), Singapore(sg), Slovenia(si), Slovakia(sk), Sierra Leone(sl), Yugoslavia(sm), Senegal(sn), Somalia(so), Spain(sp), Suriname(sr), Sao Tome & Principe(st), El Salvador(sv), Syria(sy),

Swaziland(sz), Tibet(tb), Chad(td), Togo(tg), Thailand(th), Tajikistan(tj), East Timor(tl), Turkmenistan(tm), Vietnam(tn), Tonga(to), Turkey(tr), Trinidad & Tobago(tt), Tuvalu(tv), Taiwan(tw), Tanzania(tz), Ukraine(ua), Uganda(ug), United Nations(un),United States of America(us), Uruguay(uy), Uzbekistan(uz), Vatican City(va), Saint Vincent & the Grenadines(vc), Venezuela(ve), Vietnam(vn), Vanuatu(vu), Samoa(ws), South Africa(za), Zambia(zm), Zimbabwe(zw).

The images of these flags are below:


bt.png
bw.png
by.png
bz.png
ca.png
cb.png
cf.png
cg.png
ch.png
ci.png
ck.png
cl.png
cm.png
cn.png
co.png
cr.png
ct.png
cu.png
cv.png
cx.png
cy.png
cz.png
de.png
dj.png
dk.png
dm.png
do.png
ec.png


ee.png
eg.png
eh.png
er.png
et.png
fi.png
fj.png
fr.png
ga.png
gb.png
gd.png
ge.png
gh.png
gm.png
gn.png
gq.png
gr.png
gt.png
gw.png
gy.png
hk.png
hn.png
ht.png
hu.png
id.png
ie.png
il.png
in.png


iq.png
ir.png
is.png
it.png
jm.png
jo.png
jp.png
ke.png
kg.png
kh.png
ki.png
km.png
kn.png
ko.png
kp.png
kr.png
kw.png
kz.png
la.png
lb.png
lc.png
le.png
li.png
lk.png
lr.png
lt.png
lu.png
lv.png
ly.png
ma.png
md.png
mg.png
mh.png
mi.png
mk.png
ml.png
mm.png
mn.png
mo.png
mr.png
mt.png
mu.png
mv.png
mw.png
mx.png
my.png
mz.png
na.png
ng.png
ni.png
nl.png
no.png
np.png
nr.png
nz.png
om.png
pa.png
pe.png
pg.png
ph.png
pk.png
pl.png
pt.png
pw.png
py.png
qa.png
rs.png
ru.png
rw.png
sa.png
sb.png
sc.png
sd.png
se.png
sg.png
si.png
sk.png
sl.png
sm.png
sn.png
so.png
sp.png
sr.png
st.png

Fig 1

After implementation of previous work that mostly is [6] proposed by A Lodh, we can see that actually total 65 numbers of flag images cannot be identified by the proposed system , when the query image is as mirror image.Those country are Algeria(aa), Andorra(ad), UAE(ae),_Australia(au),_Belgium(be),_Bahrain(bh), Benin(bj),_Brunei(bn), Bahamas(bs), Democratic Republic of the Congo(cb), Congo(cg), Cook Islands(ck), Cameroon(cm), Cuba(cu), Christmas Island Union(cx), Czech Republic(cz), The Dominican Republic(do), Western Sahara(eh), Eritrea(er), France(fr), Equatorial Guinea(gq), Guinea-Bissau(gw), Guyana(gy), Jordan(jo), Comoros(km), Saint Kitts & Nevis(kn), Kuwait(kw), Srilanka(lk), Moldova(md), Madagascar(mg), Mali(ml), Myanmar(mm), Malta(mt), Mexico(mx), Malaysia(my), Namibia(na), Nepal(np), New Zealand(nz), Oman(om), Panama(pa), Papua New Guinea(pg), Philippines(ph), Pakistan(pk), Portugal(pt), Qatar(qa), Solomon Islands(sb), Seychelles(sc), Sudan(sd), Slovenia(si),Slovakia(sk), Senegal(sn), Chad(td), Togo(tg), East Timor(tl), Turkmenistan(tm), Tonga(to), Trinidad & Tobago(tt), Taiwan(tw), Tanzania(tz), United States of America(us),Vatican City(va), Samoa(ws), Africa(za), Zambia(zm), Zimbabwe(zw).

There will be another dataset of 44 numbers of images which will be required for template matching. The dataset is as below :

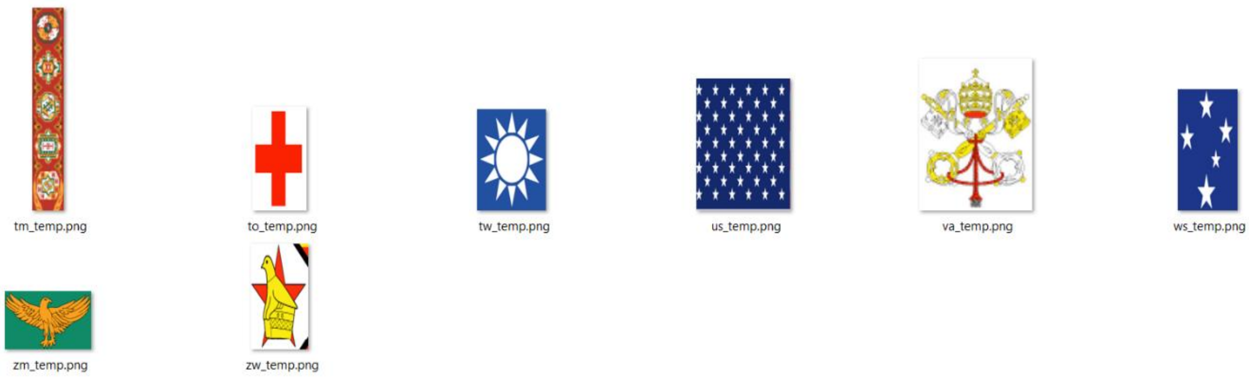


Fig 2

There will be another dataset of 21 numbers of images which will be required for HSV. The dataset is as below :

Fig 3

For checking the system another dataset of flags which are in mirror form or horizontal flipped will be used. The dataset as given below:


flip43.png
flip44.png
flip45.png
flip46.png
flip47.png
flip48.png
flip49.png
flip50.png
flip51.png
flip52.png
flip53.png
flip54.png
flip55.png
flip56.png
flip57.png
flip58.png
flip59.png
flip60.png
flip61.png
flip62.png
flip63.png
flip64.png
flip65.png
flip66.png
flip67.png
flip68.png
flip69.png
flip70.png
flip71.png
flip72.png
flip73.png
flip74.png
flip75.png
flip76.png
flip77.png
flip78.png
flip79.png
flip80.png
flip81.png
flip82.png
flip83.png
flip84.png
flip85.png
flip86.png
flip87.png
flip88.png
flip89.png
flip90.png
flip91.png
flip92.png
flip93.png
flip94.png
flip95.png
flip96.png


flip97.png
flip98.png
flip99.png
flip100.png
flip101.png
flip102.png
flip103.png
flip104.png
flip105.png
flip106.png
flip107.png
flip108.png
flip109.png
flip110.png
flip111.png
flip112.png
flip113.png
flip114.png
flip115.png
flip116.png
flip117.png
flip118.png
flip119.png
flip120.png
flip121.png
flip122.png
flip123.png
flip124.png
flip125.png
flip126.png
flip127.png
flip128.png
flip129.png
flip130.png
flip131.png
flip132.png
flip133.png
flip134.png
flip135.png
flip136.png
flip137.png
flip138.png
flip139.png
flip140.png
flip141.png
flip142.png
flip143.png
flip144.png
flip145.png
flip146.png
flip147.png
flip148.png
flip149.png
flip150.png


flip151.png
flip152.png
flip153.png
flip154.png
flip155.png
flip156.png
flip157.png
flip158.png
flip159.png
flip160.png
flip161.png
flip162.png
flip163.png
flip164.png
flip165.png
flip166.png
flip167.png
flip168.png
flip169.png
flip170.png
flip171.png
flip172.png
flip173.png
flip174.png
flip175.png
flip176.png
flip177.png
flip178.png
flip179.png
flip180.png
flip181.png
flip182.png
flip183.png
flip184.png
flip185.png
flip186.png
flip187.png
flip188.png
flip189.png
flip190.png
flip191.png
flip192.png
flip193.png
flip194.png
flip195.png
flip196.png
flip197.png
flip198.png
flip199.png
flip200.png

Fig 4

So basically, this studies' main moto is to handle those 65 numbers of flags to identify and improve the previous system.

2.2 Proposed Approach

This system will be embedded system where we will use three different methods to detect all 200 flags as original image as well as mirror image. Those methos are as following:

- Template Matching
- HSV Model
- Color Moment

.The flag pictures are thought to be noise-free since they are believed to just be synthetic images produced using photo editing software. Each image is downsized to required dimension as part of the pre-processing stage to lessen computational overloads. Additionally, white backgrounds in any photographs are eliminated to leave only the flag sections visible.

At first, this research goes for those 65 numbers of flags, which cannot be identified by the previous system. So, we start with template matching. Now here is the question. what is template matching?

2.2.1 Template matching

Template matching is nothing but a digital image processing technique that finds small parts of an image that match a template image.

A basic template matching method employs an image template tailored to a particular characteristic of the search image that we wish to detect. This technique is simple to apply to greyscale or edge images. Where the image construct and mask structure coincide, where huge image values are amplified by large mask values, the cross-correlation outcome will be at its highest. This method is typically implemented by first selecting a portion of the search image to for all intents and purposes serve as a template: The search image will be called S(x, y), in which (x, y) represent these same coordinates of each pixel inside the search image, which really is fairly significant. The template will particularly be called T(x t, y t), with (xt, yt) representing the coordinates from each pixel inside the template in a major way. Then we simply move the template's center (or origin) over every (x, y) point inside the search image and figure out the sum of products here between coefficients in S(x, y) & T(xt, yt) over entire area covered by the template. The location with the greatest score for the most part is the optimal position since all very potential placements for the template related to the search image are taken into account.

The intensity of a pixel inside the search picture having coordinates (xs, ys) is Is(xs, ys), and the intensity of a pixel inside the template having coordinates (xt, yt) is It (xt, yt). Diff(xs, ys, x t, y t) = | Is(xs, ys) - It(x t, y t) | is the definition of the absolute difference in pixel intensities.

$$SAD(x, y) = \sum_{i=0}^{T_{\text{rows}}} \sum_{j=0}^{T_{\text{cols}}} \text{Diff}(x+i, y+j, i, j)$$

The concept of looping over the pixels inside the search picture while translating the template's origin at each pixel and measuring SAD is mathematically represented as follows:

$$\sum_{x=0}^{S_{\text{rows}}} \sum_{y=0}^{S_{\text{cols}}} SAD(x, y)$$

The search image's rows and columns are denoted by Srows and Scols, whereas the template image's rows and columns are denoted by Trows and Tcols, respectively. The optimum position of the template inside the search picture is estimated using this method's lowest SAD score. Although the technique is easy to use and comprehend.

By above process all 65 numbers of flags cannot be identified properly. Only 44 numbers of flags can be identified by the above method. So, for 21 numbers of flag, another method called HSV model will be introduced.

2.2.2 HSV

HSV stand for Hue, Saturation and Value which A specifies the kind of color space. It resembles the RGB & CMYK models of today. Hue, saturation, and value are the three elements that make up the HSV color system. Sometimes the word "value" is changed to "brightness," in which case the term "HSB" is used. Alvy Ray Smith developed the HSV model in 1978. Hex-cone color model is another name for HSV.

Hue is the model's color component, given as a value from zero to 360 degrees.

The color red ranges from 0 to 60 degrees.

The color yellow ranges from 60 to 120 degrees.

The color green ranges from 120 to 180 degrees.

The color cyan ranges from 180 to 240 degrees.

The color blue ranges from 240 to 300 degrees.

The color magenta ranges from 300 to 360 degrees.

In the color space, **saturation** reveals the spectrum of grey. It falls between 0 and 100%. On occasion, the value is determined between 0 and 1. Whenever the value is "0," the color is grey; when it is "1,," a primary color, it is. A color that has faded has a reduced saturation level, which implies it has greyer in it.

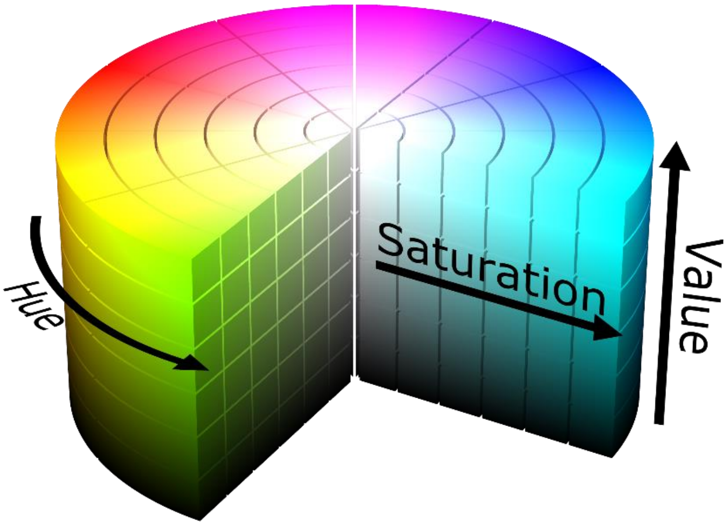


Fig 5

Value, which fluctuates with color saturation, refers to a color's brightness. It falls between 0 and 100%. The color space will be entirely black whenever the value is "0." The color space becomes brighter and displays more colors as the value rises.

HSV applications

High-quality computer graphics are frequently produced using the HSV color system. Simply said, it's used to choose the many colors that are required for a certain picture. The chosen color is chosen using an HSV color wheel. The user may utilize the color wheel to choose the specific color that is required for the image. It provides color based on how people perceive color.

Representations of HSV

To select the required color, utilize the HSV color wheel. The circle on the wheel stands for hue. Value and saturation are each represented by a different triangle. The triangle's vertical axis denotes saturation, while its horizontal axis denotes value. When choosing a color for your drawing, you must first choose a hue (the circular area), and then you may choose the required saturation from the triangle's vertical angle. You may choose the appropriate value for brightness from the triangle's horizontal angle.

HSV model is occasionally shown as a cylinder or cone. Hue is expressed by the circular portion of the cone when it is a conical object. Typically, a three-dimensional representation of the cone is used. The cone's radius is used to compute saturation, while the cone's height is used to determine value. The HSV model can alternatively be shown as a hexagonal cone. The conical model has the benefit of being able to embody the HSV color space inside a single item. Because computer interfaces are two-dimensional, the cone-shaped model of HSV is most suited for picking colors for computer graphics.

The conical model's use is comparable to that of the HSV color space's cylindrical model. Similar calculations are carried out. The cylinder type model is the HSV color space computation method that is, theoretically, the most accurate. When the value is decreased, it is practically impossible to tell the difference between saturation and hue. Due to this, the cone form is chosen over the cylindrical type, which has become obsolete.

Transformation between HSV and RGB

$\boldsymbol{H} \in [\mathbf{0}, \mathbf{360}]$

$\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B} \in [\mathbf{0}, \mathbf{1}]$

From RGB to HSV

Let MAX represent the highest (R, G, and B) value, and MIN represent the lowest.

$$H = \begin{cases} \text{undefined}, & \text{if } MAX = MIN \\ 60^\circ \times \frac{G-B}{MAX-MIN} + 0^\circ, & \text{if } MAX = R \\ & \text{and } G \geq B \\ 60^\circ \times \frac{G-B}{MAX-MIN} + 360^\circ, & \text{if } MAX = R \\ & \text{and } G < B \\ 60^\circ \times \frac{B-R}{MAX-MIN} + 120^\circ, & \text{if } MAX = G \\ 60^\circ \times \frac{R-G}{MAX-MIN} + 240^\circ, & \text{if } MAX = B \end{cases}$$

$$S = \begin{cases} 0, & \text{if } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{otherwise} \end{cases}$$

$$V = MAX$$

From HSV to RGB

$$H_i = \left\lfloor \frac{H}{60} \right\rfloor \mod 6$$

$$f = \frac{H}{60} - H_i$$

$$p = V(1 - S)$$

$$q = V(1 - fS)$$

$$t = V(1 - (1 - f)S)$$

$$\text{if } H_i = 0 \rightarrow \quad R = V, \quad G = t, \quad B = p$$

$$\text{if } H_i = 1 \rightarrow \quad R = q, \quad G = V, \quad B = p$$

$$\text{if } H_i = 2 \rightarrow \quad R = p, \quad G = V, \quad B = t$$

$$\text{if } H_i = 3 \rightarrow \quad R = p, \quad G = q, \quad B = V$$

$$\text{if } H_i = 4 \rightarrow \quad R = t, \quad G = p, \quad B = V$$

$$\text{if } H_i = 5 \rightarrow \quad R = V, \quad G = p, \quad B = q$$

In computer graphics, a number from 0 to 255 can occasionally replace a real number to represent every HSV and RGB parameter. In this instance, some distortion is brought on by rounding and the transformations do not fully cover the all points inside the target space. For instance, Both HSV value (0, 255, 255) and the HSV value (1, 255, 255) are translated to RGB value (255, 0, 0) and (255, 6, 0),

2.2.3 Color Moments

Color moments are metrics that characterize the color distribution of an image in the same manner that central moments define a probability distribution uniquely. Color moments are mostly employed in information extraction applications as features in order to assess how similar two pictures are based on color. In order to discover and obtain a comparable image, one image is often matched to a collection of digital images having pre-computed attributes. Each picture comparison generates a similarity, and the lower the number, the more similar the two photos are meant to be. Color moments are invariant to scaling and rotation. Because the majority of the color distribution information is stored in the low-order moments, only the first three color moments are often employed as attributes in image retrieval applications. Color moments are an excellent feature to employ under changing lighting circumstances since they contain both shape and color information, but they struggle with occlusion.

Color moments for any color model may be calculated. Each channel has three color moments calculated (e.g., 9 moments if the color model is RGB & 12 moments if color model is CMYK). Color moments are computed in the same way as probability distribution moments are computed.

Mean: The first color moment can be read as the image's average color, and it can be determined using the method below.

$E_i = \sum_{j=1}^{N} \frac{1}{N} p_{ij}$, where N is picture's pixel count and Pij is value of the pixel in the jth color channel of the image.

Standard Deviation: The second color moment is standard deviation, which is calculated by calculating the square root of the color distribution's variance.

$$\sigma_i = \sqrt{(\frac{1}{N} \sum_{j=1}^{N} (p_{ij} - E_i)^2)}$$

where Ei is mean value, or the first color moment, for the image's i-th color channel.

Skewness: The skewness is the third color moment. It determines how asymmetric the color distribution is and so provides information about its shape. Skewness may be calculated using the formula:

$$s_i = \sqrt[3]{(\frac{1}{N} \sum_{j=1}^{N} (p_{ij} - E_i)^3)}$$

The picture is divided into non-overlapping rectangular blocks in order to localize the color distribution in the first stage of color feature. A picture with dimension M rows and N columns is vertically subdivided into H units and vertically divided into V units, with the dimensions m by n provided by,

m = M/H , and n = N/V

The exact value of H and V is determined by experimental results. The color information for every cell I is divided into three major channels: R, G, and B. The standard deviation and mean for each channel are determined, and all six statistical characteristics are merged to describe cell information.

$$C_i = \{\mu(R), \sigma(R), \mu(G), \sigma(G), \mu(B), \sigma(B)\}$$

A combination of all such cells represents a whole flag picture.

$$I = i = 1H * VC_i$$

Each flag image's color feature thus has 6HV components.

The square root of the total of the squared distinctions between two vectors is used to determine the **Euclidean distance**. When doing distance computation hundreds or thousands of times, this is typical to eliminate the square root procedure to allow faster the calculation.

The equation for Euclidean Distance is as follows:

$$d = ((p_1 - q_1)^2 + (p_2 - q_2)^2)^{1/2}$$

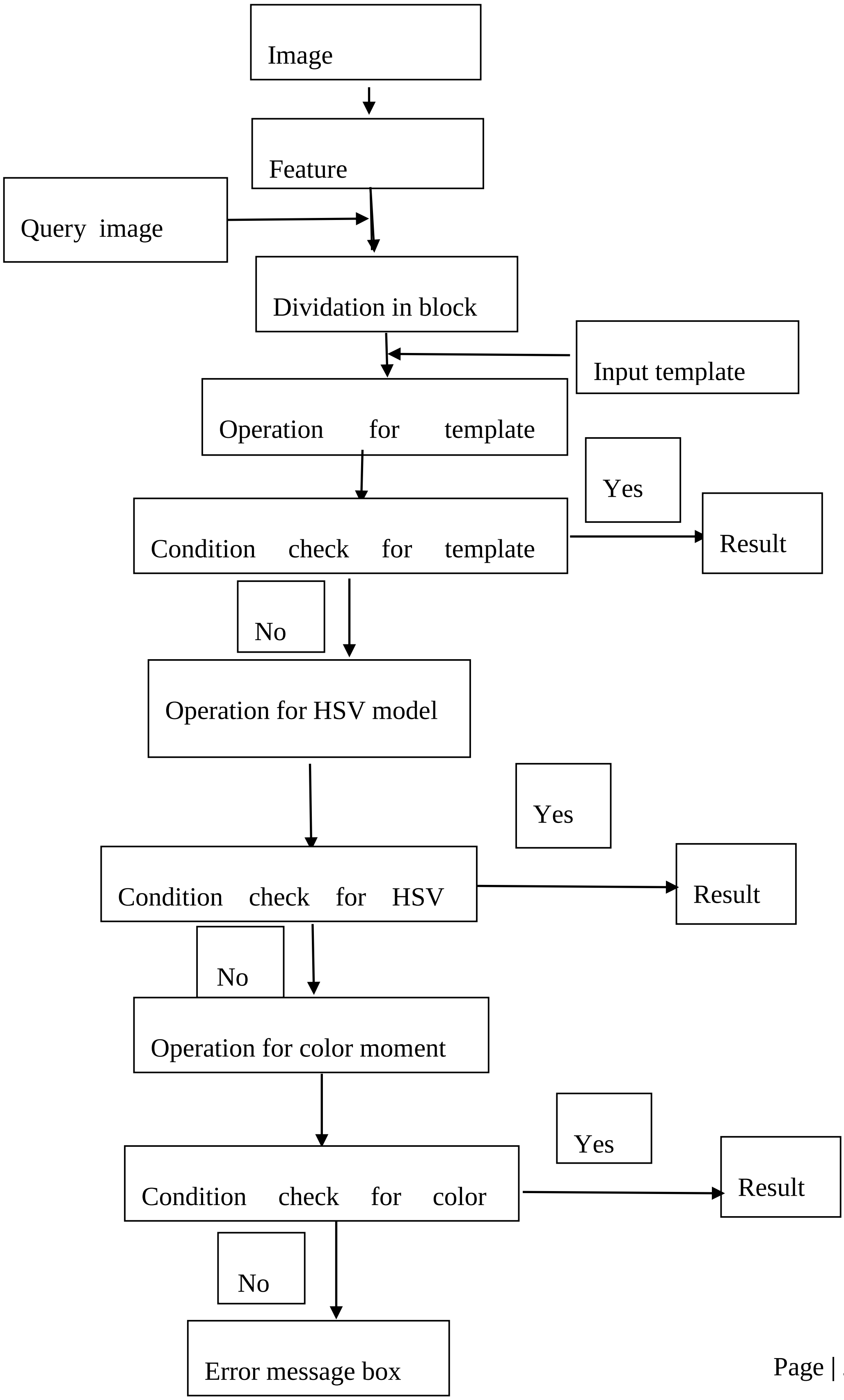
When we are working with two dimensions, we apply this formula. For an n-dimensional space, we may generalize this as follows:

$$D_e = \left(\sum_{i=1}^{n} (p_i - q_i)^2 \right)^{1/2}$$

Where, n is number of dimension and pi and qi is data point.

2.2.4 System flowchart

Figure shows a block diagram of the proposed approach's main operational blocks and data flow pathways



Chapter 3: IMPLEMENTATION & RESULT

This study will be implemented in MATLAB software. Here the implementation will be discussed along with results also be shown as step by step with MATLAB code.

In first step all operation will be done for **template matching**.

At first all 200 number of flags image will be acquired from Image database, which will be store as “training_images”, “nc” will store the length of dataset which is 200.

Next number of features will be set from each flag images as “nf” which is 72

Now the feature vector of training database will be store as “T”.

Feature vector for input image or query image will be store as Q

```
training_images=dir('.\train\*.png');

nc=length(training_images);  %no. of class

nf=72;      %no. of features for each flag

T=zeros(nc,nf);   %feature vector of training database

S=zeros(nc,nf);   %feature vector of testing database

Q=zeros(1,nf);
```

Now a for loop will be done for every image for processing the training set. Where “b” is every image. Name will be given for every image as “name” and ultimately the image name will store as “image”. Then the image will be resized as 300 by 300 pixels.

```
for b=1:length(training_images)

   name=strcat('.\train\',training_images(b).name);

   image=imread(name);
```

originalImg1=imresize(image,[400,300]); image_process; T(b,:)=F;

end

Now time to input the query image. The code for input a image shown below:

[path,~]=imgetfile();

image=imread(path);

the image will store as "image".

Now the image will be resized by 300 X 300 pixel and store as "originalImg"

originalImg=imresize(image,[300,300]);

Now the "originalImg" can be shown as below

300 X 300

Fig 6

Now the dimension of the image will be store as an array of row, column and number of color bands.

[rows columns numberOfColorBands] = size(originalImg);

In the next step the image will be divided into blocks as a certain manner. For different flag the divination will be done as basis of where the template is situated. As only 44 numbers of flags will be operated for template matching. Out of 44 numbers of flags some flags will be divided as 3 columns and 1 row, some flags will be divided as 2 columns and 1 row. Out of 44 flags total 12 numbers of flags will be divided as 2 columns and 1 row and rest 32 numbers of flag will be divided as 3 columns and 1 row.

Fig 7

fig 8

fig 9

Fig 10

fig 11

fig 12

Fig 13

fig 14

fig 15

fig 16

fig 17

Fig 18

Here the template fig 13 is part of fig 7 which is situated in left part if fig 7 divided as 2 columns and 1 row. As well as fig 14 is part of fig 8 and also the fig 8 will be divided as 2 column and the template (fig 14) is situated at right side of fig 8. And fig 9, fig 10will divided as 3 columns and 1 row. Because the templates which are fig 15, fig 16 are situated in 1st column of fig 9, 2nd column of fig 10 respectively. Whereas the templates which are fig 17 and fig 19 which are part of fig 11 and fig 12 respectively. But here templates are not belonging any specific 1st , 2nd or 3rd column. The fig 17 is belonging in both 1st and 2nd column of fig 11 adjointly. And fig 18 is belonging in both 2nd and 3rd columns of fig 12 adjointly.

32 numbers of flags will be divided as 3 columns and 1 row, which are “Domenican republic of congo”, “Jordan”, “Comoros”, “Mayanmar”, “Monaco”, “Namebia”, “Oman”, “philippins”, “Qatar”, “Solomon Island”, “Slovania”, “Timor”, “Turkmenistan”, “Tonga”, “Taiwan”, “Zimbabwe”, “Algeria”, “Andorra”, “Brunei”, “Cameroon”, “Christmas Island”, “Domenican Republic”, “Westran Sahara”, “Equatorial Guinea”, “Moldova”, “Mexico”, “Senegal”, “Baharin”, “Portugal”, “Samoa”, “Srilanka”, “Pakistan”.

The code for divide these flag images is as below:

blockSizeR = 300; %Rows in block

blockSizeC = 100; %Columns in block

Another 12 numbers of flags will be divided as 2 columns and 1 row, which are “Australia”, “Eritrea”, “Malaysia”, “Nepal”, “Panama”, “Papua New Guinea”, “Slovakia”, “United State of America”, “Cook Island”, “New zeland”, “Vatican City”, “Zambia”.

The code for divide these flag images is as below:

blockSizeR = 300; %Rows in block

blockSizeC = 150; %Columns in block

Next the size of each block in row and column will be computed and store as WholeBlockRows and WholeBlockColumns respectively. Also, the vector of each block in r row and column will be computed and store as Block respectivelyVectorR and BlockVectorC respectively.

```
%Figure out the size of each block in rows.

wholeBlockRows = floor(rows / blockSizeR);

blockVectorR = [blockSizeR * ones(1, wholeBlockRows)];

% Figure out the size of each block in columns.

wholeBlockCols = floor(columns / blockSizeC);

blockVectorC = [blockSizeC * ones(1, wholeBlockCols)];
```

Now Based on Image type a cell array as "ca" will be created which is holding some another array. If the number of Color Band is greater than 1 then ca will hold array of originalImg, blockVectorR, blockVectorC and numberOfColorBand. And if the number of Color Band is greater than 1 then ca will hold array of originalImg, blockVectorR and blockVectorC.

```
if numberOfColorBands > 1

  ca = mat2cell(originalImg, blockVectorR, blockVectorC, numberOfColorBands);

else

  ca = mat2cell(originalImg, blockVectorR, blockVectorC);

end
```

For displaying all blocks, from R, G, B values, the numerical array out of the cell array will be extracted and color moments will be calculated.

```
plotIndex = 1;

numPlotsR = size(ca, 1);

numPlotsC = size(ca, 2);

rgbBlock=cell(3,3)

for r = 1 : numPlotsR

  for c = 1 : numPlotsC

      subplot(numPlotsR, numPlotsC, plotIndex);

    rgbBlock{r,c} = ca{r,c};

    R{plotIndex}=rgbBlock{r,c}(:,:,1);  % Red

    G{plotIndex}=rgbBlock{r,c}(:,:,2);  % Green

    B{plotIndex}=rgbBlock{r,c}(:,:,3);  % Blue

    [rowsB columnsB numberOfColorBandsB] = size(rgbBlock{r,c});

    plotIndex = plotIndex + 1;

  end

end
```

Now out of numbers of flags those will be divided as 3 columns and 1 row, only “Pakistan” and “Srilanka” have the template is in last two RGB blocks, so an array which denoted by A will be sorted.

```
A = [rgbBlock{1,2},rgbBlock{1,3}];
```

Similarly, “Baharin”, “Portugal”& “Samoa” have the template is in first two RGB blocks.

```
A = [rgbBlock{1,1},rgbBlock{1,2}];
```

“Domenican republic of congo”, “Jordan”, “Comoros”, “Mayanmar”, “Monaco”, “Namebia”, “Oman”, “philippins”, “Qatar”, “Solomon Island”, “Slovania”, “Timor”, “Turkmenistan”, “Tonga”, “Taiwan”& “Zimbabwe” have the template is in first RGB blocks.

A = [rgbBlock{1,1];

“Algeria”, “Andorra”, “Brunei”, “Cameroon”, “Christmas Island”, “Domenican Republic”, “Westran Sahara”, “Equatorial Guinea”, “Moldova”, “Mexico” & “Senegal” have the template is in second RGB blocks.

A = [rgbBlock{1,2];

And out of 12 numbers of flags those will be divided as 2 columns and 1 row , 8 numbers of flags which are “Australia”, “Eritrea”, “Malaysia”, “Nepal”, “Panama”, “Papua New Guinea”, “Slovakia” & “United State of America” have the template is in first RGB blocks.

A = [rgbBlock{1,1];

And rest 4 numbers of flags which are “Cook Island”, “New zeland”, “Vatican City” & “Zambia” have the template is in second RGB blocks.

A = [rgbBlock{1,2];

In the next step the template images will be called from the data base for further operations and will be store as smallSubImage. Hence an example is given

smallSubImage = imread('path of template image of template database');

In the next step Matrix variable correlationOutput will contain the correlation co-efficient .The matrix of template and A's normalised cross-correlation are computed using the function normaxcorr2 by changing both template image and A, from RGB to gray.

correlationOutput = normxcorr2(double(rgb2gray(smallSubImage)), double(rgb2gray(A)));

Now to find the where the normalized cross-correlation in A is brightest, an array of maximum correlation value and maximum index will be created. And this will satisfy if the max correlation is higher than 0.8 . That will show in Output.

```
if maxCorrValue > 0.8
    [ypeak, xpeak] = ind2sub(size(correlationOutput),maxIndex(1));
    corr_offset = [(xpeak-size(smallSubImage,2)) (ypeak-size(smallSubImage,1))];
  end
  subplot(1, 2, 2);
  imshow(A);
```

So, if the condition that is maxCorrvalue is higher than 0.8 is satisfied then the template image will be found in input image and a message box will show the name of the country name of input flag image.

```
if maxCorrValue > 0.8
    hold on;
    rectangle('position',[corr_offset(1) corr_offset(2) size(smallSubImage,2) size(smallSubImage,1)],...
        'edgecolor','g','linewidth',2);
    title('Template Image Found in Original Image', 'FontSize', fontSize);
    msgbox('The country name of the inputted flag image')
```

And the above operation will be satisfied if and only the input image is from those 44 numbers of flag images and those images have to be original image. Now if the input image from those 44 images but in mirror form then the above system can't be satisfied. So, to satisfy the condition the next step is needed. In the next step the input image will be horizontally flipped and stored as originalImg and the rest of the step will be followed as above steps, shown the result.

```
originalImg = flip(originalImg,2);
```

But if the input image is not from those 44 numbers of flag images then the input image will go for next segment of this system which is made to serve 21 number of images by HSV model.

In this new segment number of features of input image will be reduced from 72 to 4 and rest of the features will be changed depends on nf.

```
nc=length(training_images);
nf=4;
T=zeros(nc,nf);   %feature vector of training database
S=zeros(nc,nf);   %feature vector of testing database
```

Q=zeros(1,nf); %feature vector of input image

originalImg=imresize(image,[300,300]);

[rows columns numberOfColorBands] = size(originalImg); % get dimensions of the image

In next step the input image will be divided 2 X 2 block.

blockSizeR = 150; %Rows in block

blockSizeC = 150; %Columns in block

Then the size of each block in rows and columns will be calculated:

wholeBlockRows = floor(rows / blockSizeR);

blockVectorR = [blockSizeR * ones(1, wholeBlockRows)];

wholeBlockCols = floor(columns / blockSizeC);

blockVectorC = [blockSizeC * ones(1, wholeBlockCols)];

Now Based on Image type a cell array as “ca” will be created which is holding some another array. If the number of Color Band is greater than 1 then ca will hold array of originalImg, blockVectorR, blockVectorC and numberOfColorBand. And if the number of Color Band is greater than 1 then ca will hold array of originalImg, blockVectorR and blockVectorC.

if numberOfColorBands > 1

ca = mat2cell(originalImg, blockVectorR, blockVectorC, numberOfColorBands);

else

ca = mat2cell(originalImg, blockVectorR, blockVectorC);

end

Now for each block the HSV value will be calculated by transforming the RGB to HSV. Also, the mean HSV value of those block will be store as FQ.

```
plotIndex = 1;
   numPlotsR = size(ca, 1);
   numPlotsC = size(ca, 2);
   for r = 1 : numPlotsR
      for c = 1 : numPlotsC
      rgbBlock = ca{r,c};
      HSV_image =  (rgb2hsv(rgbBlock));
      Hue{plotIndex} = HSV_image(:,:,1);
      Hue_mean(plotIndex) = mean2(Hue{plotIndex});
      figure(1),
      subplot(wholeBlockCols,wholeBlockRows,plotIndex);
      imshow(rgbBlock);
      [rowsB, columnsB, numberOfColorBandsB] = size(rgbBlock);
      plotIndex = plotIndex + 1;
      end
   end
FQ = [Hue_mean];
   Q(1,:)=FQ;
```

Now, to find the result the mean HSV value of each block of 21 number of flag image have to be known.

So, the mean HSV value of each block of those 21 numbers of flags are given below.

	Q1	Q2	Q3	Q4
United Arab Emirates	0.1684599285	0.2864239942	0.02483950813	0
Belgium	0.05295701324	0.08542519796	0.05295701324	0.08542519796
Benin	0.3237834665	0.150284093	0.2811052425	0.02264877663
Bahamas	0.2767694419	0.4130783542	0.2576756	0.3974627424
Republic of the congo	0.3883405409	0.1281169089	0.2286996379	0.03344474712
Chile	0.4141323639	0.0002035369102	0.02338915349	0.02340340652
Cuba	0.2531409839	0.3772666532	0.2380970857	0.3815186649
Czechia	0.2874604798	0	0.3182048519	0.01431914145
France	0.3882711798	0.6561888427	0.3882711798	0.6561888427
Guinea-Bissau	0.06401637646	0.151379845	0.1482546083	0.4092617137
Guyana	0.1469290157	0.3379983832	0.1519473883	0.3408762901
Saint kitts and Nevis	0.3588912926	0.0749119116	0.07579575401	0.03416465806

Kuwait	0.1708911908	0.2233719909	0.003648606273	0.00479574858
Madagascar	0.01414992874	0.02266666667	0.1383058841	0.4067656958
Seychelles	0.3596157543	0.03202055007	0.2011482896	0.2187764814
Sudan	0.1636429661	0.02110115232	0.1445918811	0
Chad	0.4666791137	0.05404899384	0.4666791137	0.05404899384
Togo	0.1089610445	0.3047963129	0.250070522	0.3046534666
Trinidad and Tobago	0.0147469877	0.02246040077	0.02261455737	0.01474748527
Tanzania	0.3963800401	0.2198796778	0.2176826312	0.5449285437
South Africa	0.2456852147	0.1466851621	0.3571997696	0.5297538537

Table 1

Now, if the mean HSV value of input image's each block is similar as those 21 images' each blocks' mean HSV value then the result will be shown in message box.

Q(1,1) >=.15 && Q(1,1) <.18 && Q(1,2) >= .27 && Q(1,2) < .29 && Q(1,3) >=.01 && Q(1,3) <.03 && Q(1,4) >=0 && Q(1,4) < .01

msgbox(''The country name of the input flag image ')

And the above operation will be satisfied if and only the input image is from those 21 numbers of flag images and those images have to be original image. Now if the input image from those 21 images but in mirror form then the above system can’t be satisfied. So, to satisfy the condition the next step is needed. In the next step the input image will be horizontally flipped and stored as originalImg and the rest of the step will be followed as above steps, shown the result.

originalImg = flip(originalImg,2);

But if the input image also is not from those 21 numbers of flag images then the input image will go for next segment of this system which is made to serve rest number of images by color moment.

In this new segment number of features of input image will be increased from 4 to 72 again and rest of the features will be changed depends on nf.

nc=length(training_images); %no. of class

nf=72; %no. of features for each flag

T=zeros(nc,nf); %feature vector of training database

S=zeros(nc,nf); %feature vector of testing database

Q=zeros(1,nf); %feature vector of input image

There will be a list of all country and their Euclidean distance will be measured .

```
d=zeros(nc,nc);
for originalImg1=1:nc
   for j=1:nc
      sum=0;
      for k=1:nf
         diff=T(originalImg1,k)-S(j,k);
         sum=sum+power(diff,2);
      end
      d(originalImg1,j)=sqrt(sum);
   end
end
```

Now Euclidean distance of input image will be measured.

```
i_d=zeros(nc,1)
for j=1:nc
      sum=0;
      for k=1:nf
```

```
        diff=T(j,k)-Q(k);

        sum=sum+power(diff,2);

    end

    i_d(j,1)=sqrt(sum);

end

fprintf('euclidean distance of input image feature vector:');

disp(i_d);
```

Now the minimum Euclidian distance will be measured between the training images and input image. Also, between all flag maximum of minimum distance will be also measured . And if the MQ is less than equal to maximum of the minimum distance then the particular country flag which have minimum distance is the result otherwise the input image is not part of all 200 flags or the image is not a flag.

```
MQ=min(i_d);

fprintf('Minimum Euclidean distance:');

disp(MQ);

if MQ <= maximum of minimum Euclidean distance

fprintf('Flag Belongs to the country :');


for originalImg1=1:length(country)
```

```
    figure(2),
if MQ==i_d(originalImg1),disp(country(originalImg1)),subplot(2,2,2),
file=training_images(originalImg1).name;
    Image_Loc = fullfile( '.\train\', file );
    imshow(Image_Loc),title(country(originalImg1));
    break;
end
end
else fprintf('The input image is not in Database or the input image is not a flag');
    msgbox('The input image is not in Database or the input image is not a flag')
end
```

Now results of all segment will be shown as below.

For the 1st segment which is for template matching there will be 6 types of different step so some all type of result is as below.

Result 1. 3 column X 1 row and template is in last 2 block.

Input image is

Original image Mirror image

Fig 19

Fig 20

Out put image

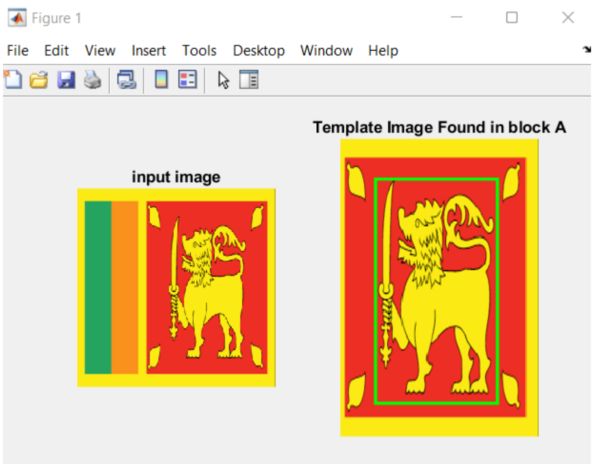


Fig 21

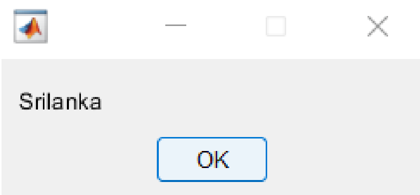


Fig 22

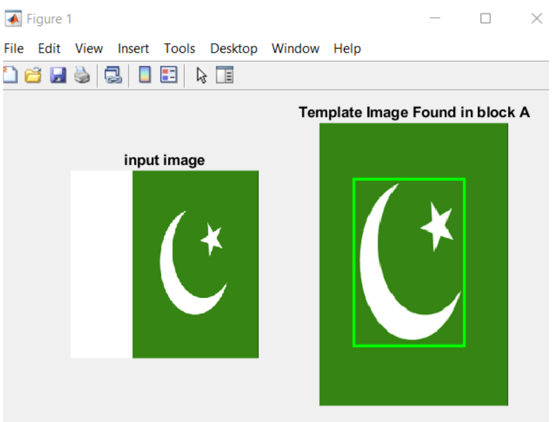


Fig 23

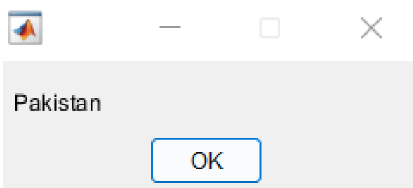


Fig 24

Result 2. 3 column X 1 row and template is in last 2 block.

In put image is

Fig 25

Fig 26

Output image

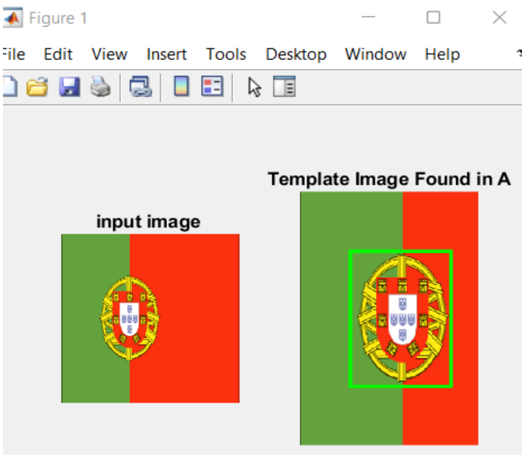


Fig 27

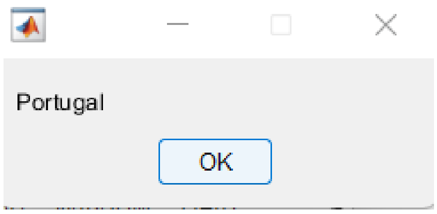


Fig 28

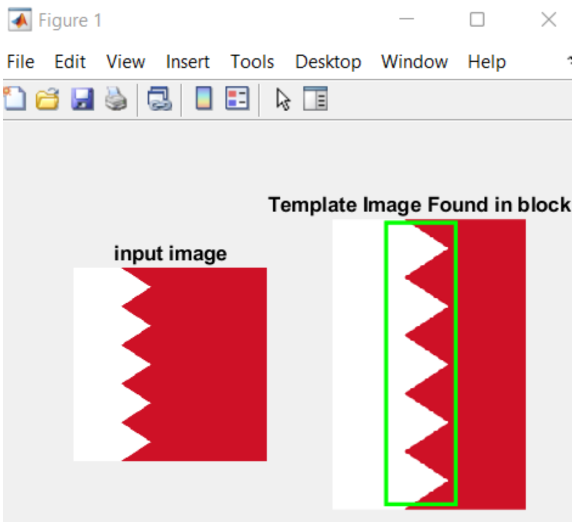


Fig 29

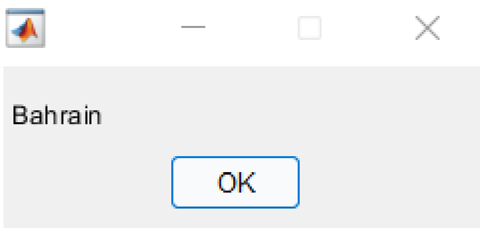


Fig 30

Result 3. 3 column X 1 row and template is in last 1st block and for mirror image template is in 2nd block.

Input image

Original image

Fig 31

Mirror image

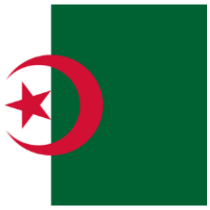


Fig 32

Output

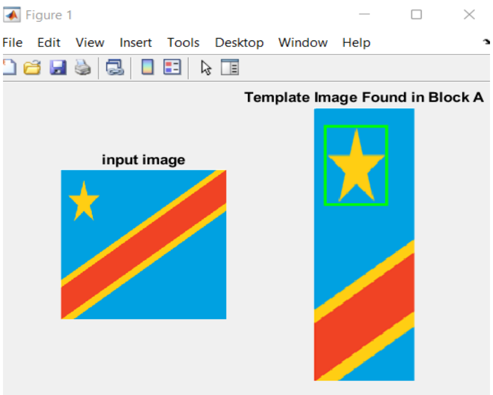


Fig 33

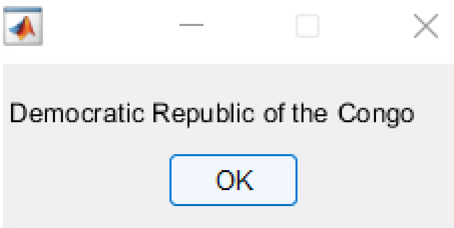


Fig 34

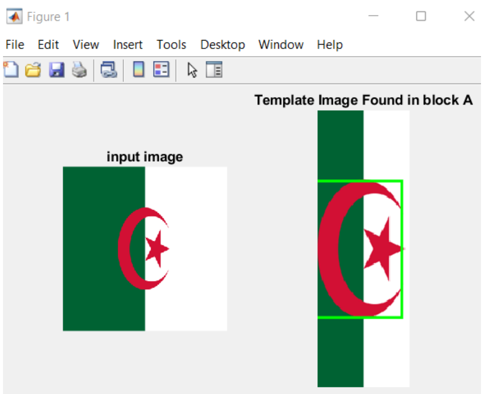


Fig 35

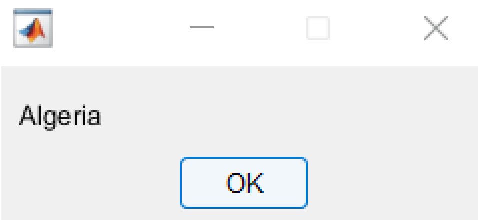


Fig 36

Result 4. 2 column X 1 row and template is in last 1st block and for mirror image template is in 2nd block.

Input image

Original image

Mirror image

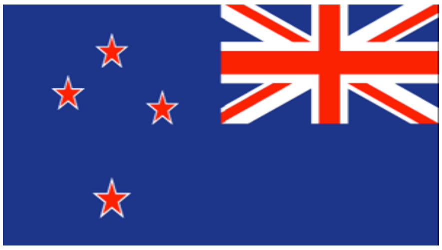
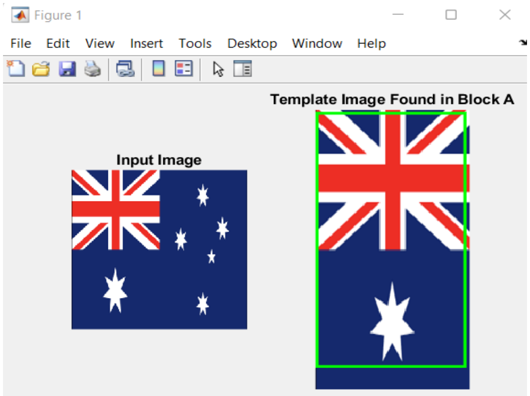


Fig 37 Fig 38

Output



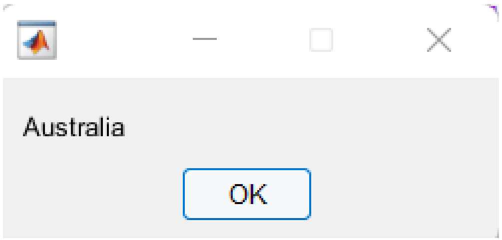
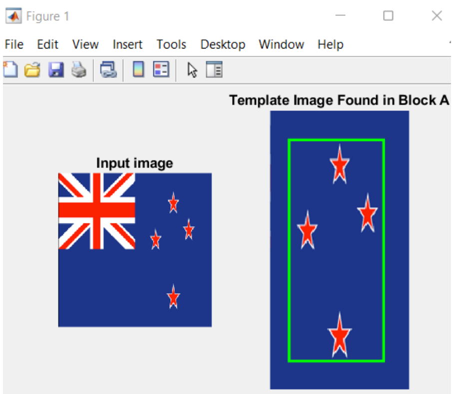


Fig 39 Fig 40



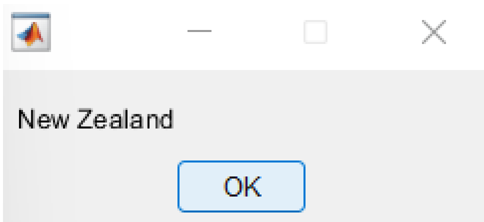


Fig 41 Fig 42

Result 5. For HSV model

Input image

Original image

Fig 43

Mirror image

Fig 44

Output image

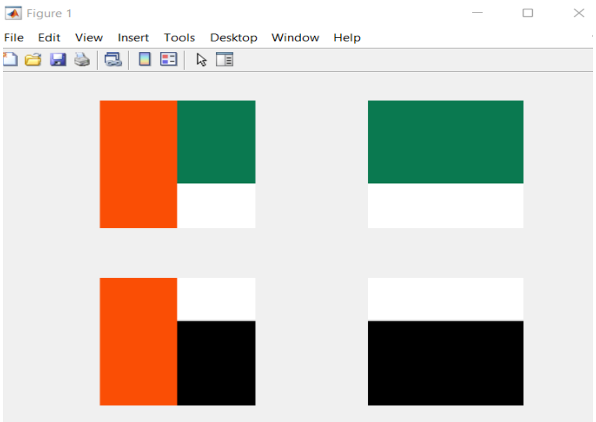


Fig 45

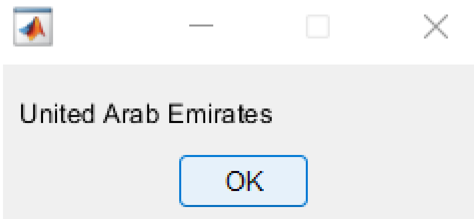


Fig 46

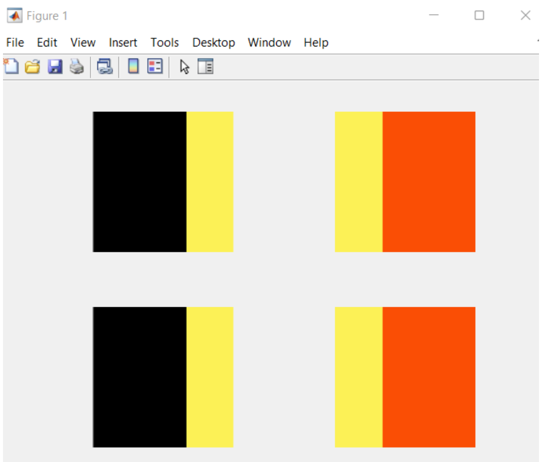


Fig 47

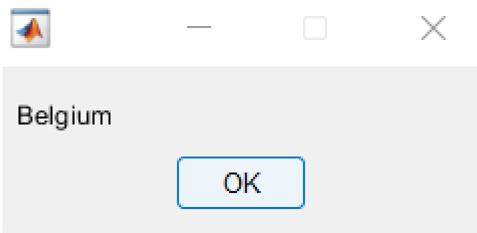


Fig 48

Result 5. Color moment

Input image

Original image mirror image

Fig 49

Fig 50

Output image

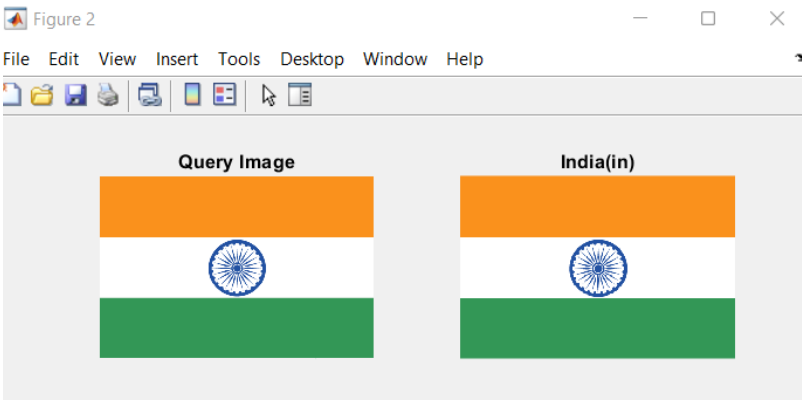


Fig 51

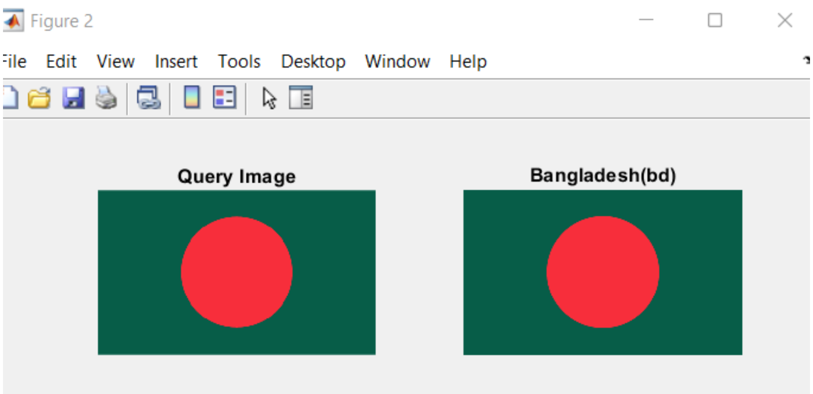


Fig 52

Result 5. Input image not in database or it's not a flag

Input image

Fig 53

Output image

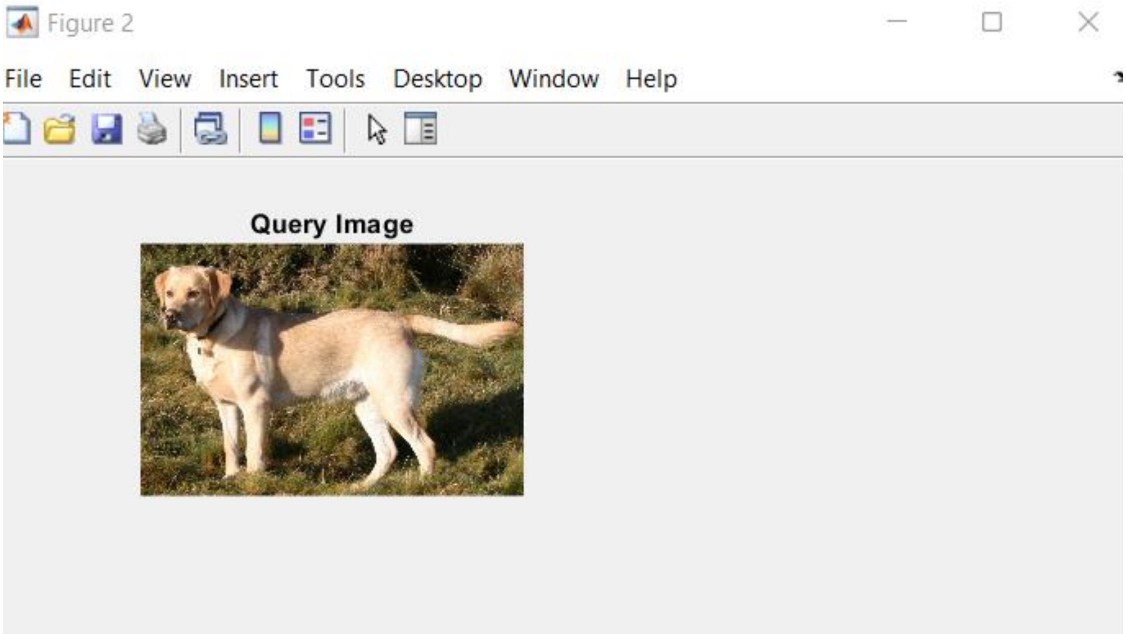


Fig 54

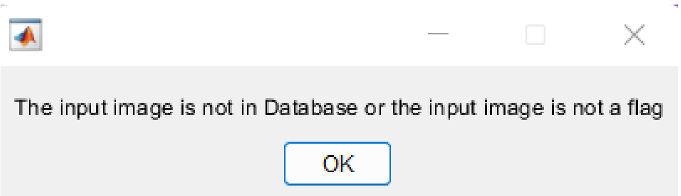


Fig 56

Chapter 4: CONCLUSION AND FUTURE SCOPE

4.1 Conclusion

This technology or system offer an advance way to detect 200 countries' flag more efficiently whether it is as original or mirror image. As there are total 3 advanced image processing systems has been used so chances of error is lesser .

In this system has also some limitation. First, this system can't identify a flag when the input image is in any type of rotated form. Second, there will be arise problems if the color intensity of input images is far different from database images.

4.2 Future Scope

Already this system is enough efficient to detect or identify 200 flags. But there are lot of scopes to upgrade this this system. This system can be upgraded to detect rotated flags. This system also can be upgraded to prevent the 2nd limitation which is discussed in conclusion.

Chapter 5: REFERENCES

[1]Lodh, Avishikta & Parekh, Ranjan, “Computer Aided Identification of Flags using Color Features”. *International Journal of Computer Applications.* 149. 1-7. 10.5120/ijca2016911587. 2016

[2]S. Jetley, A. Vaze, S. Belhe, “Automatic Flag Recognition Using Texture Based Color Analysis andGradient Features”, Proceedings of the 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), pp. 464-469, 2013.

[3]E. Hart, S. H. Cha, C. Tappert, "Interactive Flag Identification Using a Fuzzy-Neural Technique", *Proceedings of Student/Faculty Research Day, CSIS, Pace University*, May 7th, 2004.

[4]Foschi G. Patricia, “Feature Extraction for Image Mining”, Romberg , International Workshop on Multimedia Information Systems, p 1-7

[5] Eduardo Hart, Sung-Hyuk Cha, Charles C. Tappert, “Interactive Flag Identification Using Image Retrieval Techniques”. *January 2004 Conference: Proceedings of the International Conference on Imaging Science, Systems and Technology, CISST '04,* June 21-24, 2004, Las Vegas, Nevada, USAx

[6]Md. Z. Rahman, Md. S. Hossain, K. Ahmed, “Flag Identification Using Support Vector Machine”, JU Journal of Information Technology (JIT), Vol. 2, pp. 11- 16 June 2013

[7]M. B. Asad, F. J. Sharna, N. Alam and M. J. Islam, "HSV and Template Matching based Bengali road sign recognition technique," *2016 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, 2016, pp. 1-4, doi: 10.1109/ICISET.2016.7856527.

[8]C. -H. Su, H. -S. Chiu and T. -M. Hsieh, "An efficient image retrieval based on HSV color space," *2011 International Conference on Electrical and Control Engineering*, 2011, pp. 5746-5749, doi: 10.1109/ICECENG.2011.6058026.

[9]M. A. El-Hallaq, "A Proposed Template Image Matching Algorithm for Face Recognition," *2019 IEEE 7th Palestinian International Conference on Electrical and Computer Engineering (PICECE)*, 2019, pp. 1-6, doi: 10.1109/PICECE.2019.8747239.

[10]P. Ganesan, V. Rajini, B. S. Sathish and K. B. Shaik, "HSV color space-based segmentation of region of interest in satellite images," *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2014, pp. 101-105, doi: 10.1109/ICCICCT.2014.6992938.

[11] Pitas, Ioannis. Digital image processing algorithms and applications. John Wiley & Sons, 2000.

[12] Wilber, Michael J., et al. "Animal recognition in the mojave desert: Vision tools for field biologists." *2013 IEEE Workshop on Applications of Computer Vision (WACV*). IEEE, 2013.

[13] S. Mangijao Singh , K. Hemachandran, “Image Retrieval Based on the Combination of Color Histogram and Color Moment”, *International Journal of Computer Applications, Vol.* 58, No. 3, pp. 27-34, 2012.

[14] T. V. Saikrishna, A. Yesubabu, A. Anandarao, T. Sudha Rani, “A Novel Image Retrieval Method Using Segmentation and Color Moments”, *Advanced Computing: An International Journal*, Vol. 3, No. 1, pp. 75-80 2012.

[15] G. Mandloi, “A Survey of Feature Extraction Techniquesfor Color Images”, *International Journal of Computer Science and Information Technologies*, Vol. 5, No. 3, pp.4615-4620, 2014

[16] H. H. P. K. Bhuravarjula, V. N. S. Vijaya Kumar, "ANovel Image Retrieval Method using color Moments", *International Journal Of Electronics And Computer Science Engineering (Ijecse ,ISSN:2277-1956),* Vol. 01, No. 04, pp. 2432-2438, 2012.

[17] A. Chadha, S. Mallik, R. Johar, “Comparative Study and Optimization of Feature Extraction Techniques for Content Based Image Retrieval”, *International Journal of Computer Applications*, Vol. 52, No. 20, pp. 35-42,b 2012.

[18] S. Mangijao Singh , K. Hemachandran, “Content-Based Image Retrieval Using Color Moments and Gabor Texture Feature”, *International Journal of Computer Science Issues*, Vol. 9, Issue 5, No. 1, pp. 299-308, 2012.

[19] C. Yang and X. Li, "The green fluorescence image segmentation with the combination of HSV model and watershed algorithm," *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2015, pp. 607-610, doi: 10.1109/ICSESS.2015.7339131.

[20] R. Parekh, “Principles of Multimedia*”, 2nd Edition, Published by Tata McGraw-Hill Education,* 2012

[21] X. Qi and L. Miao, "A Template Matching Method for Multi-Scale and Rotated Images Using Ring Projection Vector Conversion," *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 2018, pp. 45-49, doi: 10.1109/ICIVC.2018.8492726.

[22] Goshtasby, Arthur, Stuart H. Gage, and Jon F. Bartholic. "A two-stage cross correlation approach to template matching." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 3 : 374-378.1984

[23] Venugopal, Gayathri, et al. "Fast template matching for intra prediction." *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017.

[24] J. N. Sarvaiya, S. Patnaik and S. Bombaywala, "Image Registration by Template Matching Using Normalized Cross-Correlation," *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, 2009, pp. 819-822, doi: 10.1109/ACT.2009.207.

[25] Di Stefano, Luigi, Stefano Mattoccia, and Martino Mola. "An efficient algorithm for exhaustive template matching based on normalized cross correlation." *12th International Conference on Image Analysis and Processing, 2003. Proceedings.*. IEEE, 2003.

[26] www.wikipedia.com

[27] World Flag Database (URL : www.flags.net)

[28] Flags of the World (URL : www.crwflags.com)

[29] Flagpedia (URL : www.flagpedia.net)

[30] www.mathworks.com

Appendix :

Total 56 numbers of images and 1 number of tables have been used to finish this study.

Sources of figure:

Fig 5 : https://en.wikipedia.org/wiki/File:HSV_color_solid_cylinder_saturation_gray.png

Fig 53: https://en.wikipedia.org/wiki/File:Labrador_on_Quantock_(2175262184).jpg

Hardware:

1) Dell inspiron 15 3000

Software:

1) Matlab 21
2) Ms word
3) Ms excel