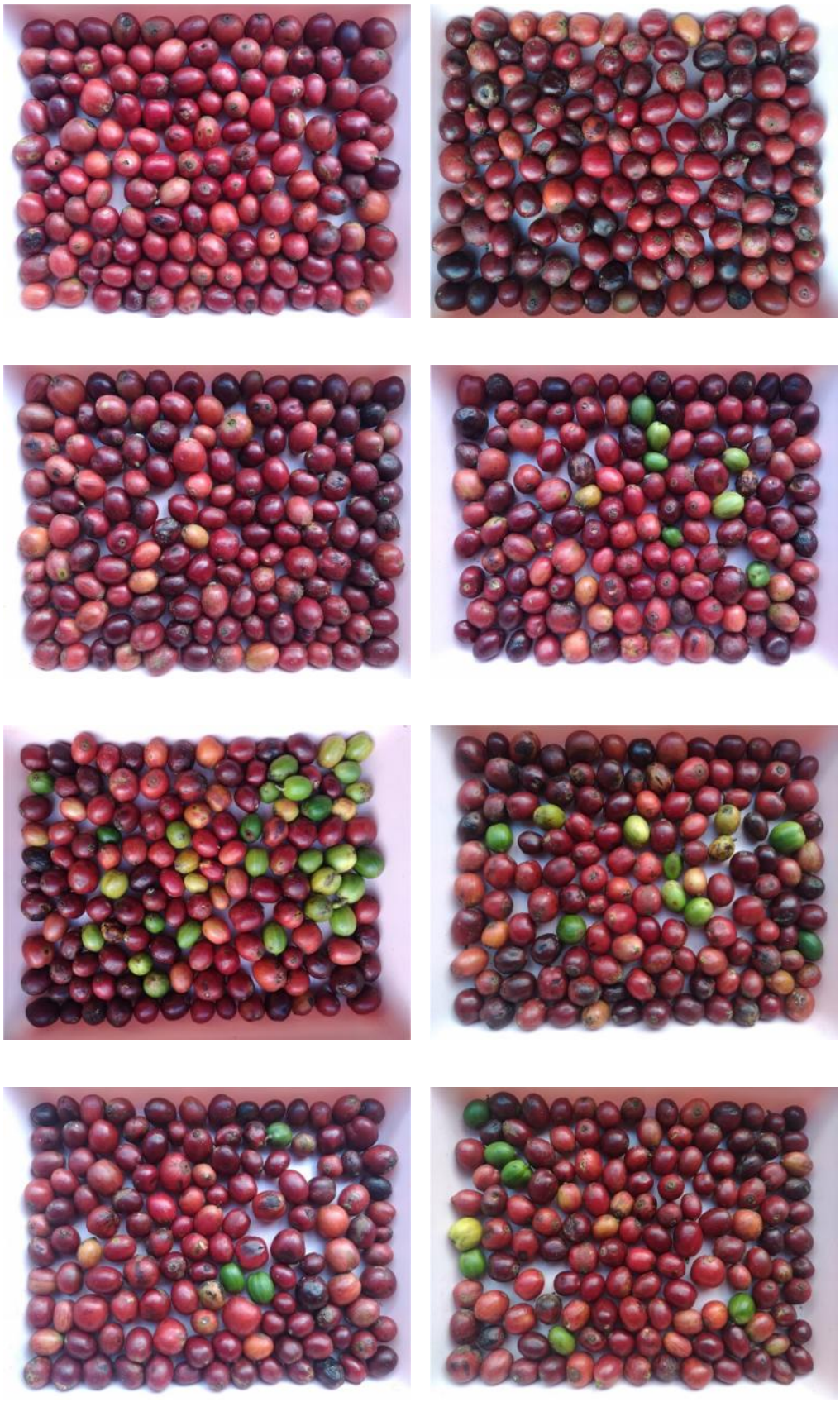
LAMPIRAN

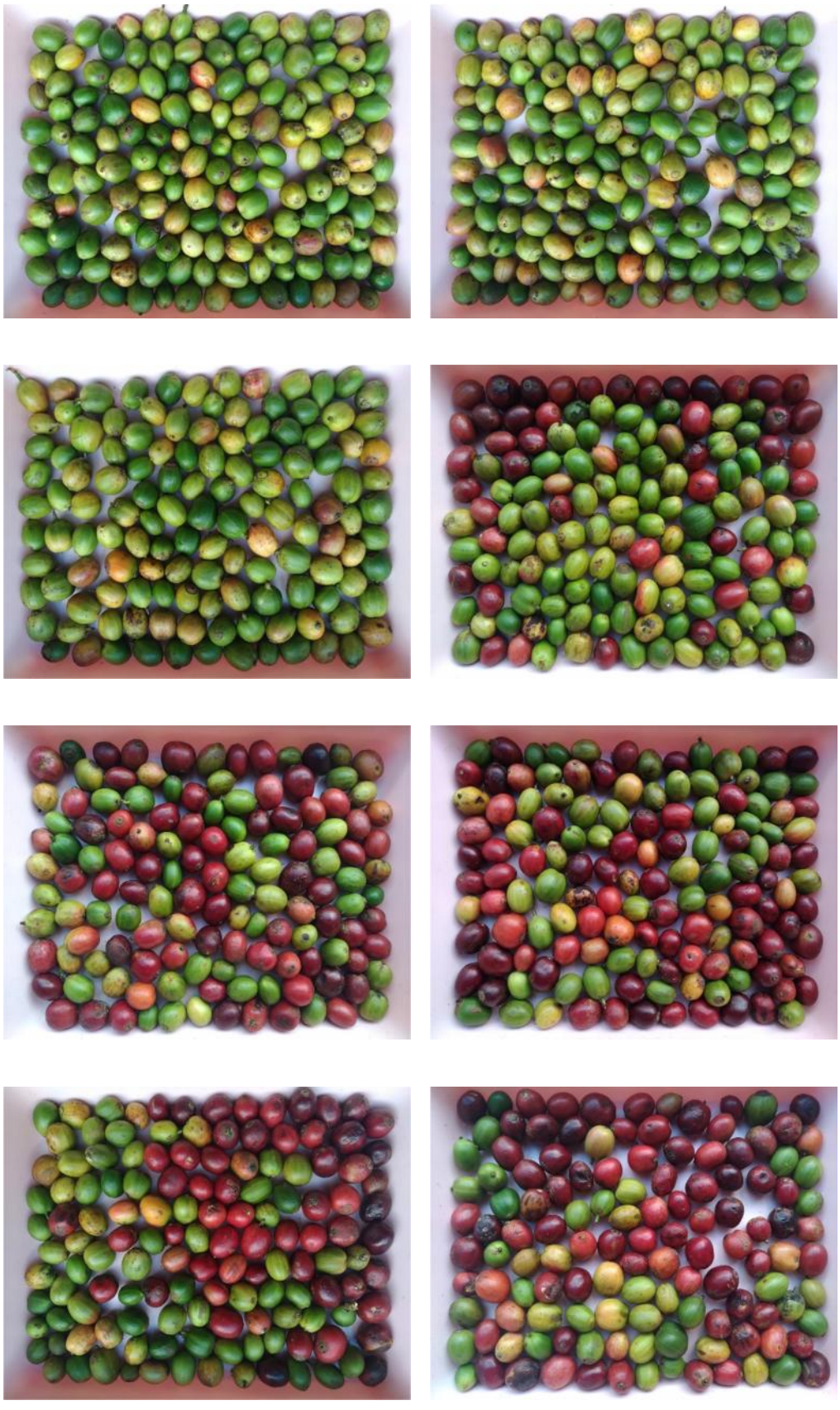
LAMPIRAN A

Citra Digital Pelatihan

Kategori 1 (Matang)

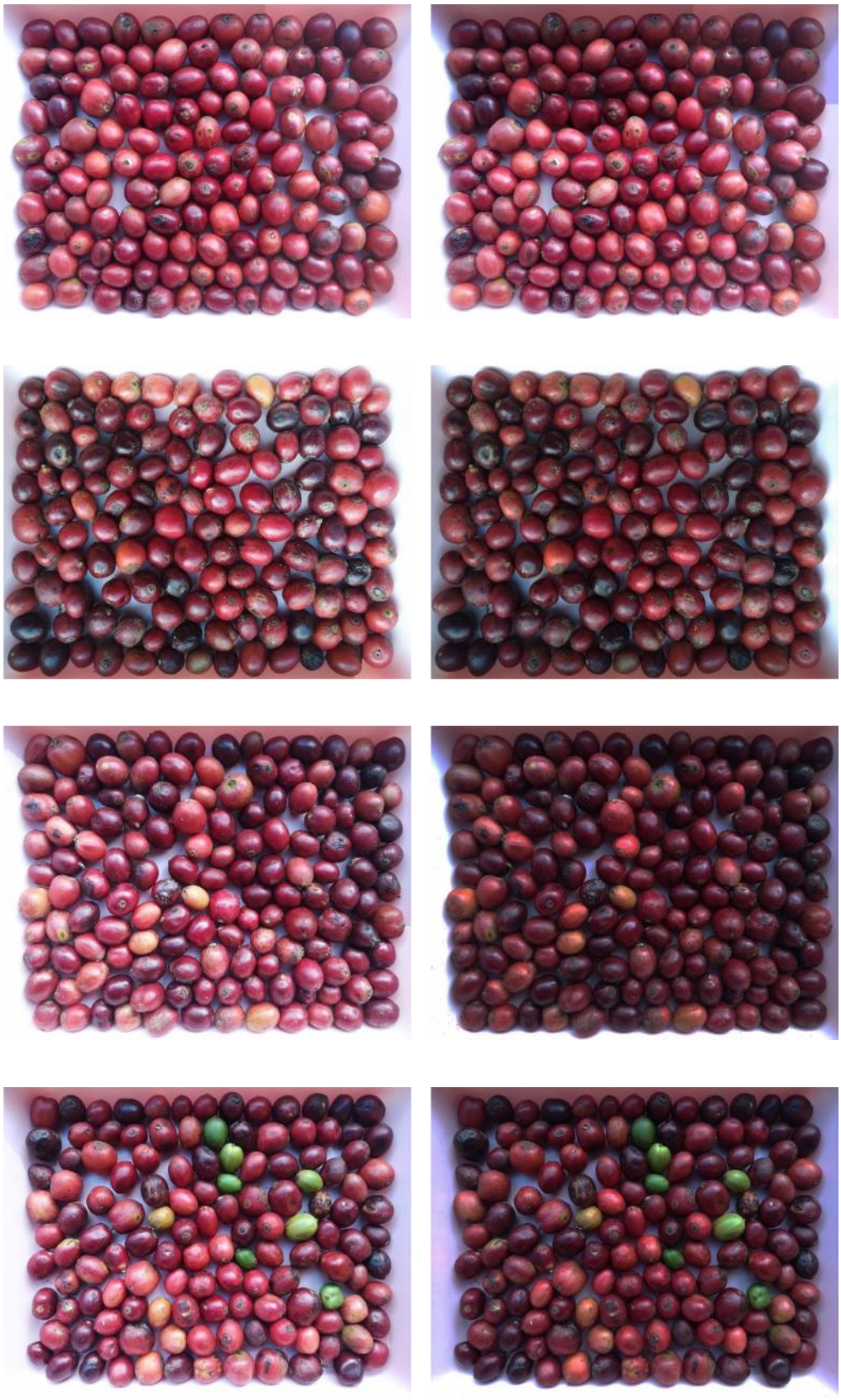


Kategori 2 (Mentah)



LAMPIRAN B

Citra Digital dengan Perubahan Intensitas Cahaya
























LAMPIRAN C

Data Pelatihan, Hasil Ekstraksi, dan Pengujian

Tabel Lampiran C.1 Hasil Ekstraksi Citra Digital

No.	Kategori 1 = Matang 2 = Mentah	Nilai Rata-rata		
		R (*Red*)	G (*Green*)	B (*Blue*)
1.	1	151.436	87.7992	108.648
2.	1	125.626	84.2045	93.9942
3.	1	137.888	86.6625	104.052
4.	1	142.886	97.3188	114.559
5.	1	134.606	92.4286	97.6189
6.	1	127.808	89.6326	94.4606
7.	1	146.566	103.691	120.838
8.	1	136.047	94.8274	99.66
9.	2	122.61	134.982	91.2794
10.	2	129.617	138.273	99.5434
11.	2	115.531	122.718	83.7302
12.	2	116.46	107.925	81.3734
13.	2	131.493	107.948	96.4894
14.	2	125.158	95.5377	86.829
15.	2	114.709	98.685	84.3805
16.	2	129.893	104.846	100.337

Tabel Lampiran C.2 Bobot Awal

No.	Bobot			Kategori
1.	0.4167	0.2549	0.3285	1
2.	0.3800	0.3394	0.1806	2

Tabel Lampiran C.3 Hasil Pengujian Black Box

No.	Keterangan	Berjalan dengan baik?
1.	Tombol *Browse*	Ya
2.	Tampilkan Gambar Hasil *Browse*	Ya
3.	Hasil Klasifikasi	Ya

LAMPIRAN D

Kode Program

Kode Program Lampiran D.1 Kode Program untuk Ekstraksi Ciri Warna RGB

```
g = imread('cd');   % Lokasi file image

red=mean(mean(g(:,:,1)));   %Mean R
green=mean(mean(g(:,:,2))); %Mean G
blue=mean(mean(g(:,:,3)));  %Mean B
```

Kode Program Lampiran D.2 Kode Program Pelatihan

```
clc; clear; close all; warning off all;
% Data latih dan target latih
data_latih = [151.436, 125.626, 137.888, 142.886,
134.606, 127.808, 146.566, 136.047, 122.61,  129.617,
115.531, 116.46,  131.493, 125.158, 114.709, 129.893;...
87.7992, 84.2045, 86.6625, 97.3188, 92.4286, 89.6326,
103.691, 94.8274, 134.982, 138.273, 122.718, 107.925,
107.948, 95.5377, 98.685,  104.846;...
108.648, 93.9942, 104.052, 114.559, 97.6189, 94.4606,
120.838, 99.66, 91.2794, 99.5434, 83.7302, 81.3734,
96.4894, 86.829,  84.3805, 100.337];

target_latih = [1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0;...
                0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1];

% Membuat fungsi NN LVQ
net = newlvq(data_latih,2); % 2 adalah jumlah klasifikasi

% Memberikan nilai untuk memengaruhi proses pelatihan
net.performFcn = 'mse';
net.trainParam.goal = 0.01;
net.trainParam.epochs = 100;
net.trainParam.lr = 0.05;

net.IW {1,1} = [0.4167, 0.2549, 0.3285;...
                0.3800, 0.3394, 0.1806];

% Proses Training
net = train(net,data_latih,target_latih);

% Bobot Akhir
bobot_akhir = net.IW {1,1}
```

Kode Program Lampiran D.3 Kode Program GUI Pengujian

```
function varargout = lvq_pengujian(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @lvq_pengujian_OpeningFcn, ...
                   'gui_OutputFcn',  @lvq_pengujian_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function lvq_pengujian_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

[back] = imread('background.jpg');
image (back)


function varargout = lvq_pengujian_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function axes3_CreateFcn(hObject, eventdata, handles)

function edit2_Callback(hObject, eventdata, handles)
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% Browse Button untuk mengambil citra dari direktori
% dan menampilkan hasil
function pushbutton1_Callback(hObject, eventdata, handles)
[nama_file,nama_path] = uigetfile(...
    {'*.bmp; *.jpg; *.*'},...
    'Buka Citra');
if ~isequal (nama_file,0)
    handles.data1 = imread(fullfile(nama_path,nama_file));
    guidata(hObject,handles);
    handles.current_data1=handles.data1;
    axes(handles.axes1);
    imshow(handles.data1);
else
    return
end

red=mean(mean(handles.data1(:,:,1)));   %Mean R
green=mean(mean(handles.data1(:,:,2))); %Mean G
blue=mean(mean(handles.data1(:,:,3)));  %Mean B

load jstlvqdataset16citra.mat

data_uji = [red;green;blue];
hasil_uji = round(sim(net,data_uji))

if hasil_uji== [1;0];
    set(handles.edit2,'string','Matang');
else
    set(handles.edit2,'string','Mentah');
end
```

Kode Program Lampiran D.4 Kode Program Perhitungan dengan Matlab

```
clc; clear; close all; warning off all;
% Data latih dan target latih
data_latih = [151.436, 125.626, 122.61,  129.617;...
87.7992, 84.2045, 134.982, 138.273;...
108.648, 93.9942, 91.2794, 99.5434];

target_latih = [1,1,0,0;...
                0,0,1,1];
net = newlvq(data_latih,2);
net.performFcn = 'mse';
net.trainParam.goal = 0.01;
net.trainParam.epochs = 1;
net.trainParam.lr = 0.05;

net.IW {1,1} = [0.4167, 0.2549, 0.3285;...
                0.3800, 0.3394, 0.1806];

% Proses Training
net = train(net,data_latih,target_latih);

% Bobot Akhir
bobot_akhir = net.IW {1,1}
```

Kode Program Lampiran D.5 Pelatihan dan Pengujian dengan Python

```
import numpy as np
import neurolab as nl

input = np.array ([[151.436, 87.7992, 108.648],
                   [125.626, 84.2045, 93.9942],
                   [137.888, 86.6625, 104.052],
                   [142.886, 97.3188, 114.559],
                   [134.606, 92.4286, 97.6189],
                   [127.808, 89.6326, 94.4606],
                   [146.566, 103.691, 120.838],
                   [136.047, 94.8274, 99.66],
                   [122.61, 134.982, 91.2794],
                   [129.617, 138.273, 99.5434],
                   [115.531, 122.718, 83.7302],
                   [116.46, 107.925, 81.3734],
                   [131.493, 107.948, 96.4894],
                   [125.158, 95.5377, 86.829],
                   [114.709, 98.685, 84.3805],
                   [129.893, 104.846, 100.337]])
target = np.array ([   [1, 0],
                       [1, 0],
                       [1, 0],
                       [1, 0],
                       [1, 0],
                       [1, 0],
                       [1, 0],
                       [1, 0],
                       [0, 1],
                       [0, 1],
                       [0, 1],
                       [0, 1],
                       [0, 1],
                       [0, 1],
                       [0, 1],
                       [0, 1]])
net = nl.net.newlvq(nl.tool.minmax(input), 4, [.8, .2])
error = net.train(input, target, show=1, epochs=100,
goal=0.001, lr=0.05)

test = net.sim ([[151.436, 87.7992, 108.648],
                   [125.626, 84.2045, 93.9942],
                   [137.888, 86.6625, 104.052],
                   [142.886, 97.3188, 114.559],
                   [134.606, 92.4286, 97.6189],
                   [127.808, 89.6326, 94.4606],
                   [146.566, 103.691, 120.838],
                   [136.047, 94.8274, 99.66],
                   [122.61, 134.982, 91.2794],
                   [129.617, 138.273, 99.5434],
                   [115.531, 122.718, 83.7302],
                   [116.46, 107.925, 81.3734],
                   [131.493, 107.948, 96.4894],
                   [125.158, 95.5377, 86.829],
                   [114.709, 98.685, 84.3805],
                   [129.893, 104.846, 100.337]])
print (test)
```