<table>
<thead>
<tr>
<th>No</th>
<th>Judul</th>
<th>Nama Seminar</th>
<th>Tempat</th>
<th>Tahun</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pemanfaatan <em>Fuzzy Logic</em> Dalam menakar Prestasi Siswa Sesuai Standar Kompetensi Sekolah</td>
<td>Konferensi Nasional Pengembangan Teknologi Informasi dan Komunikasi</td>
<td>Hotel Karibia, Medan</td>
<td>2014</td>
</tr>
<tr>
<td>2</td>
<td>Pemanfaatan <em>Fuzzy Logic</em> Dalam Prosedur Penerimaan Pegawai Baru</td>
<td>Lomba Karya Tulis Ilmiah &amp; Seminar PT. Inalum (Persero)</td>
<td>Hotel Grand Antares, Medan</td>
<td>2015</td>
</tr>
<tr>
<td>3</td>
<td>Pemanfaatan <em>Fuzzy Logic</em> Dalam Sistem Penerimaan Pegawai Baru</td>
<td>Jurnal TIMES</td>
<td>STMIK TIME, Medan</td>
<td>2016</td>
</tr>
</tbody>
</table>
LAMPIRAN SOURCE CODE PROGRAM MLVQ

1. Training MLVQ

Imports System.Data.OleDb
Imports Microsoft.VisualBasic
Imports System.IO.StreamReader
Imports System.Text
Imports System.IO

Public Class frmTrainKombi
    Dim ID As Integer
    Dim ItemList1 As New ArrayList()
    Dim ItemList2 As New ArrayList()
    Dim str1(5) As String
    Dim BinerImg As String
    Dim UserID As Integer
    Dim NmFile1 As String
    Dim mask(9) As Integer
    Dim ResizeImg As Image
    Dim NamaFileTrain, BinerTT As String
    Dim mask(9) As Integer
    Dim milseconds, seconds, minutes, hours As Integer
    Dim input1(100) As Integer
    Dim input2(100) As Integer
    Dim v(100, 100), v1(100, 100), v2(100, 100), w(100), w0, z_in(66), w1, z(66), Y_in, y_akt, delta_w(100) As Double
    Dim delta_in(100), delta(100), delta_v(100, 100), delta_v0(100), delta_w0 As Double
    Dim lamda, learning_rate As Single
    Dim Binari As Integer, IDBobot As Integer
    Dim mat_A(100, 100) As Integer
    Dim mat_B(100, 100) As Integer
    Dim mat_C(100, 100) As Integer
    Dim db_Hasil(100, 100) As Integer
    Dim mat_Input(100, 2) As Integer
    Dim mat_target(50), db_target(50), target(50), bobot1, bobot2 As Integer
    Dim temp As Integer

Private Sub frmSignUp_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Call KoneKsi()
    Call ViewData()
    Dim i As Integer = 0
    Do While (i < 9)
        mask(i) = 1
        i = (i + 1)
    Loop
btnLoad.Enabled = False
cmdSave.Enabled = False
btnProses.Enabled = False
pb.Visible = False
lv.Enabled = True
btnAdd.Enabled = True
btnAdd.Focus()
End Sub
Sub CariUserID()
    CMD = New OleDbCommand("Select * from tUser order by UserID desc", Conn)
    RD = CMD.ExecuteReader()
    RD.Read()
    If RD.HasRows = True Then
        UserID = RD.Item("UserID") + 1
    Else
        UserID = 1
    End If
    RD.Close()
    CMD.Dispose()
End Sub

Sub CariID()
    CMD = New OleDbCommand("Select * from TrainKombi order by ID desc", Conn)
    RD = CMD.ExecuteReader()
    RD.Read()
    If RD.HasRows = True Then
        ID = RD.Item("ID") + 1
    Else
        ID = 1
    End If
    RD.Close()
    CMD.Dispose()
End Sub

Sub ViewData()
    lv.Items.Clear()
    Dim lvDaf As ListViewItem
    With lv
        .Columns.Clear()
        .Columns.Add("ID Data", 50, HorizontalAlignment.Left)
        .Columns.Add("Nama", 150, HorizontalAlignment.Left)
        .View = View.Details
        .GridLines = True
        .BorderStyle = BorderStyle.Fixed3D
        .AutoArrange = True
    End With
    CMD = New OleDbCommand("select * from tUser order by UserID asc", Conn)
RD = CMD.ExecuteReader()
If RD.HasRows Then
    Do While RD.Read
        str(0) = RD.Item("UserID")
        str(1) = RD.Item("UserName")
        lvDaf = New ListViewItem(str)
        lv.Items.Add(lvDaf)
    Loop
End If
RD.Close()
CMD.Dispose()
End Sub

Private Sub btnLoad_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLoad.Click
    btnLoad.Enabled = False
    OpenFileDialog1.Filter = "File Wajah|*.bmp;*.jpg"
    OpenFileDialog1.Title = "File Wajah"
    If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
        PictureBox1.Image = Image.FromFile(OpenFileDialog1.FileName)
        lblNmFile.Text = OpenFileDialog1.FileName
        NamaFileTrain = OpenFileDialog1.FileName
        btnProses.Enabled = True
    End If
    pb.Visible = True
End Sub

Private Sub btnProses_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnProses.Click
    btnProses.Enabled = False
    Dim MaxEpoh As Integer
    Dim Epoh, max_epoh As Integer, i As Integer
    Dim alfa, target_Error, kuadrat_Error_rerata As Double
    MaxEpoh = Val(txtEpoch.Text)
    Timer1.Interval = 10
    Timer1.Start()
    For i = 0 To 3
        v(0, i) = VBMath.Rnd
    Next
    For i = 0 To 3
        v(1, i) = VBMath.Rnd
    Next
    For i = 0 To 3
        v0(i) = VBMath.Rnd
    Next
    For i = 0 To 3
        w(i) = VBMath.Rnd
    Next
    w0 = VBMath.Rnd
    alfa = txtLearningRate.Text
    max_epoh = MaxEpoh
target_Error = txtErr.Text
Epoh = 0
kuadrat_Error_rerata = 0

Call ImgProses()
Call ReadPixel()
Call Matriks_Biner()
Call Reduksi()
Call Transposisi()
Call Set_Target()
Call ModKombi()
pb.Value = 100
Timer1.Stop()
cmdSave.Enabled = True
MsgBox("Pilih tombol Save untuk menyimpan data bobot")
End Sub
Sub CariIDBobot()
CMD = New OleDbCommand("Select * from BobotKombi order by IDBobot _
desc", Conn)
RD = CMD.ExecuteReader()
RD.Read()
If RD.HasRows = True Then
  IDBobot = RD.Item("IDBobot") + 1
Else
  IDBobot = 1
End If
RD.Close()
CMD.Dispose()
End Sub
Private Sub cmdSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdSave.Click
If lblNama.Text <> "" Then
  Call CariID()
  CMD = New OleDbCommand("Insert into TrainKombi _
(ID,UserID,NmFile,Biner) values (" & ID & ",", & UserID & "," & _
  lblNmFile.Text & "," & BinerImg & ")", Conn)
  RD = CMD.ExecuteReader()
  RD.Close()
  CMD.Dispose()
  Call CariIDBobot()
  CMD = New OleDbCommand("Insert into BobotKombi
(IDBobot,ID,Epoch,LR,ER,Waktu) values (" & IDBobot & "," & ID & "," & 
  RD = CMD.ExecuteReader
  RD.Close()
  CMD.Dispose()
  Dim i As Integer
  For i = 0 To 3
CMD = New OleDbCommand("Update BobotKombi SET v1" & i + 1 & 
" = " & v(0, i) & ",v2" & i + 1 & " = " & v1(i) & ",v0" & i + 1 & " = " & v0(i) & ",w" & 
i + 1 & " = " & w(i) & ", IDBobot = " & IDBobot & ", Conn)
        RD = CMD.ExecuteReader
        RD.Close()
        CMD.Dispose()
Next
        MsgBox("Data anda sudah disimpan")
cmdSave.Enabled = False
        PictureBox1.Image = Nothing
        UserID = 0
        btnLoad.Enabled = False
        pb.Value = 0
        pb. Visible = False
        btnAddEnabled = True
        miliseconds = 0
        seconds = 0
        minutes = 0
        hours = 0
Else
        MsgBox("Lengkapi nama ")
        lblNama.Focus()
End If
End Sub

Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As _
System.EventArgs) Handles Cancel.Click
        Close()
End Sub

Sub ReadPixel()
        Dim tempbmp1 As New Bitmap(PictureBox1.Image)
        Dim Red1 As Integer, Green1 As Integer, Blue1 As Integer
        Dim X, Y As Integer, Gray1 As Integer
        With (tempbmp1)
            For X = 0 To . Height - 1
                Application.DoEvents()
                For Y = 0 To . Width - 1
                    Red1 = (. GetPixel(Y, X). R)
                    If Red1 < 0 Then
                        Red1 = 0
                    End If
                    If Red1 > 255 Then
                        Red1 = 255
                    End If
                    Green1 = (. GetPixel(Y, X). G)
                    If Green1 < 0 Then
                        Green1 = 0
                    End If
                    Blue1 = (. GetPixel(Y, X). B)
                    If Blue1 < 0 Then
                        Blue1 = 0
                    End If
                Next
            Next
        End With
End Sub
End If
If Green1 > 255 Then
    Green1 = 255
End If
Blue1 = (.GetPixel(Y, X).B)
If Blue1 < 0 Then
    Blue1 = 0
End If
If Blue1 > 255 Then
    Blue1 = 255
End If
Gray1 = (Red1 + Green1 + Blue1) / 3
Dim Tr As Integer
If Gray1 < 128 Then
    Tr = 0
Else
    Tr = 1
End If
BinerImg = BinerImg & Tr
lbTarget.Items.Add(Tr)
Next
Next
End With

End Sub

Sub ImgProses()
    pb.Value = 20
    Dim LebarOri, TinggiOri As Integer
    Dim bmp As New Bitmap(NamaFileTrain)
    LebarOri = bmp.Width.ToString()
    TinggiOri = bmp.Height.ToString()
    If LebarOri > 100 And TinggiOri > 100 Then
        LebarImg = 100
        TinggiImg = 100
        Rectangle.BackgroundImage = Image.FromFile(NamaFileTrain)
        Dim newSize As New Size(100, 100)
        ResizeImg = New Bitmap(Rectangle.BackgroundImage, newSize)
        Rectangle.BackgroundImage = ResizeImg
        Rectangle.BackgroundImageLayout = ImageLayout.Stretch
    Else
        Rectangle.BackgroundImage = Image.FromFile(NamaFileTrain)
    End If
    Dim source As New Bitmap(Rectangle.BackgroundImage)
    Dim red, green, blue, grayscale As Integer
    RectangleShape2.BackgroundImage = Nothing
End Sub
For y As Integer = 0 To source.Height - 1
    For x As Integer = 0 To source.Width - 1
        red = source.GetPixel(x, y).R
        green = source.GetPixel(x, y).G
        blue = source.GetPixel(x, y).B
        grayscale = red * 0.299 + green * 0.587 + blue * 0.114
        source.SetPixel(x, y, Color.FromArgb(grayscale, grayscale, grayscale))
    Next x
Next y
RectangleShape2.BackgroundImage = source
Dim img As Bitmap = New Bitmap(RectangleShape2.BackgroundImage)
Dim c As Color
Dim ii As Integer = 0
Do While (ii < img.Width)
    Dim jj As Integer = 0
    Do While (jj < img.Height)
        If (((ii - 1) >= 0) AndAlso ((jj - 1) >= 0)) Then
            c = img.GetPixel((ii - 1), (jj - 1))
            mask(0) = Convert.ToInt16(c.R)
        Else
            mask(0) = 0
        End If
        If (((jj - 1) >= 0) AndAlso (ii + 1 < img.Width)) Then
            c = img.GetPixel((ii + 1), (jj - 1))
            mask(1) = Convert.ToInt16(c.R)
        Else
            mask(1) = 0
        End If
        If ((jj - 1) >= 0) Then
            c = img.GetPixel(ii, (jj - 1))
            mask(2) = Convert.ToInt16(c.R)
        Else
            mask(2) = 0
        End If
        If (ii + 1 < img.Width) Then
            c = img.GetPixel((ii + 1), jj)
            mask(3) = Convert.ToInt16(c.R)
        Else
            mask(3) = 0
        End If
        If (((ii - 1) >= 0) AndAlso (jj + 1 < img.Height)) Then
            c = img.GetPixel((ii - 1), (jj + 1))
            mask(4) = Convert.ToInt16(c.R)
        Else
            mask(4) = 0
        End If
    End While
    If (((ii - 1) >= 0) AndAlso (jj + 1 < img.Height)) Then
        c = img.GetPixel((ii - 1), (jj + 1))
    Else
        c = img.GetPixel((ii - 1), (jj + 1))
    End If
    ii = ii + 1
End While
mask(5) = Convert.ToInt16(c.R)
Else
    mask(5) = 0
End If

If (jj + 1 < img.Height) Then
    c = img.GetPixel(ii, (jj + 1))
    mask(6) = Convert.ToInt16(c.R)
Else
    mask(6) = 0
End If

If ((ii + 1 < img.Width) AndAlso (jj + 1 < img.Height)) Then
    c = img.GetPixel((ii + 1), (jj + 1))
    mask(7) = Convert.ToInt16(c.R)
Else
    mask(7) = 0
End If

Dim sum As Integer = 0
Dim u As Integer = 0
Do While (u < 9)
    sum = (sum + mask(u))
    u = (u + 1)
Loop
sum = (sum / 9)
img.SetPixel(ii, jj, Color.FromArgb(sum, sum, sum))
jj = (jj + 1)
Loop

Dim source1 As New Bitmap(RectangleShape3.BackgroundImage)
Dim sobelResult As New Bitmap(RectangleShape3.BackgroundImage)
Dim sobelX, sobelY, magnitude As Integer
Dim neighbourList As ArrayList = New ArrayList
For y As Integer = 0 To source1.Height - 1
    For x As Integer = 0 To source1.Width - 1
        neighbourList.Clear()
        neighbourList = getThreeNeighbourList(x, y, source1)
        sobelX = getSobelValue(neighbourList, "X")
        sobelY = getSobelValue(neighbourList, "Y")
        If magnitude > 255 Then
            magnitude = 255
        End If
    Next
Next
End If

    sobelResult.SetPixel(x, y, Color.FromArgb(magnitude, magnitude, magnitude))
Next x
Next y

picSobel.Image = sobelResult
picSobel.SizeMode = PictureBoxSizeMode.StretchImage
PictureBox1.Image = picSobel.Image

End Sub

Public Function getThreeNeighbourList(ByVal xPos As Integer, ByVal yPos As Integer, ByVal source As Bitmap) As ArrayList
    Dim neighbourList As ArrayList = New ArrayList
    Dim xStart, xFinish, yStart, yFinish As Integer
    Dim pixel As Integer
    xStart = xPos - 1
    xFinish = xPos + 1
    yStart = yPos - 1
    yFinish = yPos + 1
    For y As Integer = yStart To yFinish
        For x As Integer = xStart To xFinish
            If (x < 0) Or (y < 0) Or (x > source.Width - 1) Or (y > source.Height - 1) Then
                neighbourList.Add(0)
            Else
                pixel = source.GetPixel(x, y).R
                neighbourList.Add(pixel)
            End If
        Next x
    Next y
    Return neighbourList
End Function

Public Function getSobelValue(ByVal neighbourList As ArrayList, ByVal maskType As String) As Integer
    Dim result As Integer = 0
    Dim sobelX As Integer(,) = {{-1, 0, 1}, {-2, 0, 2}, {-1, 0, 1}}
    Dim sobelY As Integer(,) = {{1, 2, 1}, {0, 0, 0}, {-1, -2, -1}}
    Dim count As Integer = 0
    If (maskType.Equals("X")) Then
        For y As Integer = 0 To 2
            For x As Integer = 0 To 2
                result = result + (sobelX(x, y) * Convert.ToInt16(neighbourList(count)))
                count = count + 1
            Next x
        Next y
    ElseIf (maskType.Equals("Y")) Then

For y As Integer = 0 To 2
    For x As Integer = 0 To 2
        result = result + (sobelY(x, y) * 
            Convert.ToInt16(neighbourList(count)))
        count = count + 1
    Next x
Next y
End If

Return result
End Function

Private Sub txtClear_Click(ByVal sender As System.Object, ByVal e As 
    System.EventArgs) Handles txtClear.Click
    PictureBox1.Image = Nothing
    Rectangle.BackgroundImage = Nothing
    RectangleShape2.BackgroundImage = Nothing
    RectangleShape3.BackgroundImage = Nothing
    picSobel.Image = Nothing
    lblNama.Text = ""
    lblNmFile.Text = ""
    NamaFileTrain = ""
    lblNama.Text = ""
    NmPemilik = ""
    ID = 0
    hours = 0
End Sub

Private Sub lv_Click(ByVal sender As Object, ByVal e As 
    System.EventArgs) Handles lv.Click
If bolTambah = True Then
    UserID = lv.FocusedItem.SubItems(0).Text
    CMD = New OleDbCommand("select * from tUser where UserID=" & _ 
        UserID & ", Conn)
    RD = CMD.ExecuteReader()
    If RD.Read Then
        lblNama.Text = RD.Item("UserName")
        lblNmFile.Text = RD.Item("NmFile")
        btnLoad.Enabled = True
    Else
        NmPemilik = ""
        ID = 0
        lblNama.Text = ""
        lblNmFile.Text = ""
    End If
    RD.Close()
End Sub
CMD.Dispose()
End If
End Sub

Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAdd.Click
bolTambah = True
bolCari = False
btnAdd.Enabled = False
lv.Enabled = True
End Sub

Sub Matriks_Biner()
Dim i As Integer, j As Integer
Dim temp As Integer
Dim Y As Integer
Dim X As Integer
Dim tempbmp1 As New Bitmap(picSobel.Image)
Dim Red As Integer, Green As Integer, Blue As Integer, Grey As Integer
Dim AppendTo As Boolean
Dim strBin As String
With tempbmp1
For Y = 0 To 99 'picSobel.Height
    i = Y
    For X = 0 To 99 'picSobel.Width
        Application.DoEvents()
        j = X
        Red = CInt(.GetPixel(X, Y).R)
        Green = CInt(.GetPixel(X, Y).G)
        Blue = CInt(.GetPixel(X, Y).B)
        Grey = (76 * Red) / 255 + 150 * Green / 255 + 28 * Blue / 255
        If Grey > 190 Then
            Binari = 0
        Else
            Binari = 1
        End If
        lbTarget.Items.Add(Binari)
        BinerTT = BinerTT & Binari
        mat_A(i, j) = Binari
        If (X = 0 And Y = 0) Then
            strBin = Binari
        Else
            If (X = 100 And Y = 100) Then
                strBin = Binari
            Else
                strBin = strBin & " " & Binari
            End If
        End If
    End If
End Sub
Try
    AppendTo = False
Catch ex As Exception
    MsgBox(ex.Message, MsgBoxStyle.Exclamation, "Save _
    Error")
End Try
Next
Next
End With
X = 0

For Y = 0 To 99
    i = Y
    For X = 0 To 99
        Application.DoEvents()
        j = X
        temp = mat_A(i, j)
        Next X
    Next Y
    pb.Value = 65
End Sub

Sub Reduksi()
    Dim i As Integer, j As Integer, k As Integer, l As Integer, m As Integer
    Dim Hasil As Integer
    Dim n As Integer
    Dim X As Integer, Y As Integer
    i = 0
    m = 0
    l = 0
    For k = 0 To 9
        Application.DoEvents()
        m = l + 9
        For i = 1 To m
            For j = 0 To 9
                n = mat_A(i, j)
                Hasil = Hasil + n
                mat_B(k, 0) = Hasil
            Next j
            Next i
            Hasil = 0
            For i = 1 To m
                For j = 10 To 19
                    n = mat_A(i, j)
                    Hasil = Hasil + n
                    mat_B(k, 1) = Hasil
                Next j
                Next i
            Hasil = 0
            For i = 1 To m
                For j = 10 To 19
                    n = mat_A(i, j)
                    Hasil = Hasil + n
                    mat_B(k, 1) = Hasil
                Next j
                Next i
        End For
    End For
For i = 1 To m
    For j = 20 To 29
        n = mat_A(i, j)
        Hasil = Hasil + n
        mat_B(k, 2) = Hasil
    Next j
    Next i

Hasil = 0
For i = 1 To m
    For j = 30 To 39
        n = mat_A(i, j)
        Hasil = Hasil + n
        mat_B(k, 3) = Hasil
    Next j
    Next i

Hasil = 0
For i = 1 To m
    For j = 40 To 49
        n = mat_A(i, j)
        Hasil = Hasil + n
        mat_B(k, 4) = Hasil
    Next j
    Next i

Hasil = 0
For i = 1 To m
    For j = 50 To 59
        n = mat_A(i, j)
        Hasil = Hasil + n
        mat_B(k, 5) = Hasil
    Next j
    Next i

Hasil = 0
For i = 1 To m
    For j = 60 To 69
        n = mat_A(i, j)
        Hasil = Hasil + n
        mat_B(k, 6) = Hasil
    Next j
    Next i

Hasil = 0
For i = 1 To m
    For j = 70 To 79
        n = mat_A(i, j)
        Hasil = Hasil + n
        mat_B(k, 7) = Hasil
    Next j
    Next i
Hasil = Hasil + n
mat_B(k, 7) = Hasil
Next j
Next i
Hasil = 0

For i = 1 To m
  For j = 80 To 89
    n = mat_A(i, j)
    Hasil = Hasil + n
    mat_B(k, 8) = Hasil
  Next j
Next i
Hasil = 0

l = l + 10
Next k

For Y = 0 To 9
  i = Y
  For X = 0 To 9
    j = X
    temp = mat_B(i, j)
    Next X
  Next Y
End Sub

Sub Transposisi()
  Dim i As Integer, j As Integer
  Dim X As Integer, Y As Integer
  Dim temp, temp1 As Integer
  Application.DoEvents()

  For Y = 0 To 1
    For X = 0 To 9
      temp = mat_B(Y, X)
      mat_C(X, Y) = temp
      Next X
    Next Y
  For Y = 2 To 3
    For X = 0 To 9
      temp = mat_B(Y, X)
      mat_C(X + 10, Y - 2) = temp
      Next X
  Next Y
End Sub
Next Y

For Y = 4 To 5
    For X = 0 To 9
        temp = mat_B(Y, X)
        mat_C(X + 20, Y - 4) = temp
    Next X
Next Y

For Y = 6 To 7
    For X = 0 To 9
        temp = mat_B(Y, X)
        mat_C(X + 30, Y - 6) = temp
    Next X
Next Y

For Y = 8 To 9
    For X = 0 To 9
        temp = mat_B(Y, X)
        mat_C(X + 40, Y - 8) = temp
    Next X
Next Y

For Y = 0 To 49
    i = Y
    For X = 0 To 1
        j = X
        temp = mat_C(i, j)
        If (temp > 0) And (temp < 100) Then
            temp1 = 1
            mat_Input(i, j) = temp1
        Else
            temp1 = 0
            mat_Input(i, j) = temp1
        End If
    Next X
Next Y

pb.Value = 85
End Sub

Sub Set_Target()
    Dim temp1, temp2, temp3, Y As Integer
    For Y = 0 To 49
        temp1 = mat_Input(Y, 0)
        temp2 = mat_Input(Y, 1)
        temp3 = temp1 Or temp2
        mat_target(Y) = lbTarget.Items(Y).ToString
    Next Y
End Sub

Sub ModKombi()
    Dim temp1, temp2, temp3, Epoh, max_epoh, i, j As Integer
Dim alfa, target_Error, per_Error, kuadrat_Error, delta_Error, kuadrat_Error_rerata As Double
Dim MaxEpoh As Integer
Dim mat_Input(100, 100) As Integer
Dim mat_Target(100), db_Target(100), Target(100) As Integer
MaxEpoh = Val(txtEpoch.Text)
For i = 0 To MaxEpoh
    temp1 = mat_Input(i, 0)
    input1(i) = temp1
    temp2 = mat_Input(i, 1)
    input2(i) = temp2
    temp3 = mat_Target(i)
    Target(i) = temp3
Next i
For i = 0 To 3
    v(0, i) = VBMath.Rnd
Next
For i = 0 To 3
    v(1, i) = VBMath.Rnd
Next
For i = 0 To 3
    v0(i) = VBMath.Rnd
Next
w0 = VBMath.Rnd
alfa = txtLearningRate.Text
max_epoh = MaxEpoh
target_Error = txtErr.Text
Epoh = 0
kuadrat_Error_rerata = 0
Do
    Epoh = Epoh + 1
    For j = 0 To Val(txtEpoch.Text)
        For i = 0 To 2
            z_in(i) = v0(i) + (v(0, i) * input1(j)) + (v(1, i) * input2(j))
        Next i
        i = 0
        For i = 0 To 2
            z(i) = 1 / (1 + Math.Exp(-1 * z_in(i)))
            z(i) = Math.Round(z(i), 4)
        Next i
        i = 0
        Y_in = w0 + (w(0) * z(0)) + (w(1) * z(1)) + (w(2) * z(2)) + (w(3) * z(3))
        y_akt = 1 / (1 + Math.Exp(-1 * Y_in))
        y_akt = Math.Round(y_akt, 4)
    Next j
    Epoh = Epoh + 1
    For j = 0 To Val(txtEpoch.Text)
        For i = 0 To 2
            z_in(i) = v0(i) + (v(0, i) * input1(j)) + (v(1, i) * input2(j))
        Next i
        i = 0
        For i = 0 To 2
            z(i) = 1 / (1 + Math.Exp(-1 * z_in(i)))
            z(i) = Math.Round(z(i), 4)
        Next i
        i = 0
        Y_in = w0 + (w(0) * z(0)) + (w(1) * z(1)) + (w(2) * z(2)) + (w(3) * z(3))
        y_akt = 1 / (1 + Math.Exp(-1 * Y_in))
        y_akt = Math.Round(y_akt, 4)
    Next j
    Epoh = Epoh + 1
    For j = 0 To Val(txtEpoch.Text)
        For i = 0 To 2
            z_in(i) = v0(i) + (v(0, i) * input1(j)) + (v(1, i) * input2(j))
        Next i
        i = 0
        For i = 0 To 2
            z(i) = 1 / (1 + Math.Exp(-1 * z_in(i)))
            z(i) = Math.Round(z(i), 4)
        Next i
        i = 0
        Y_in = w0 + (w(0) * z(0)) + (w(1) * z(1)) + (w(2) * z(2)) + (w(3) * z(3))
        y_akt = 1 / (1 + Math.Exp(-1 * Y_in))
        y_akt = Math.Round(y_akt, 4)
    Next j
    Epoh = Epoh + 1
    For j = 0 To Val(txtEpoch.Text)
        For i = 0 To 2
            z_in(i) = v0(i) + (v(0, i) * input1(j)) + (v(1, i) * input2(j))
        Next i
        i = 0
        For i = 0 To 2
            z(i) = 1 / (1 + Math.Exp(-1 * z_in(i)))
            z(i) = Math.Round(z(i), 4)
        Next i
        i = 0
        Y_in = w0 + (w(0) * z(0)) + (w(1) * z(1)) + (w(2) * z(2)) + (w(3) * z(3))
        y_akt = 1 / (1 + Math.Exp(-1 * Y_in))
        y_akt = Math.Round(y_akt, 4)
    Next j
    Epoh = Epoh + 1
    For j = 0 To Val(txtEpoch.Text)
        For i = 0 To 2
            z_in(i) = v0(i) + (v(0, i) * input1(j)) + (v(1, i) * input2(j))
        Next i
        i = 0
        For i = 0 To 2
            z(i) = 1 / (1 + Math.Exp(-1 * z_in(i)))
            z(i) = Math.Round(z(i), 4)
        Next i
        i = 0
        Y_in = w0 + (w(0) * z(0)) + (w(1) * z(1)) + (w(2) * z(2)) + (w(3) * z(3))
        y_akt = 1 / (1 + Math.Exp(-1 * Y_in))
        y_akt = Math.Round(y_akt, 4)
    Next j
    Epoh = Epoh + 1
per_Error = Target(j) - y_akt
kuadrat_Error = per_Error ^ 2
kuadrat_Error = Math.Round(kuadrat_Error, 4)
kuadrat_Error_rerata = kuadrat_Error_rerata + kuadrat_Error
delta_Error = per_Error * y_akt * (1 - y_akt)

For i = 0 To 2
    delta_w(i) = alfa * delta_Error * z(i)
    delta_w(i) = Math.Round(delta_w(i), 5)
Next i

i = 0
delta_w0 = alfa * delta_Error

For i = 0 To 2
    delta_in(i) = delta_Error * w(i)
    delta_in(i) = Math.Round(delta_in(i), 5)
Next i

i = 0
For i = 0 To 2
    delta(i) = delta_in(i) * (1 / (1 + Math.Exp(-1 * z(i)))) * (1 - (1 / (1 + Math.Exp(-1 * z(i)))))
    delta(i) = Math.Round(delta(i), 5)
Next i

i = 0
For i = 0 To 2
    Application.DoEvents()
    delta_v(0, i) = alfa * delta(i) * input1(j)
    delta_v(0, i) = Math.Round(delta_v(0, i), 5)
    delta_v(1, i) = alfa * delta(i) * input2(j)
    delta_v(1, i) = Math.Round(delta_v(1, i), 5)
    delta_v0(i) = alfa * delta(i)
    delta_v0(i) = Math.Round(delta_v0(i), 5)
Next i

i = 0
For i = 0 To 2
    Application.DoEvents()
    v(0, i) = v(0, i) + delta_v(0, i)
    v(0, i) = Math.Round(v(0, i), 5)
    v1(0, i) = v(0, i)
    v(1, i) = v(1, i) + delta_v(1, i)
    v(1, i) = Math.Round(v(1, i), 5)
    v2(1, i) = v(1, i)
    v0(i) = v0(i) + delta_v0(i)
    v0(i) = Math.Round(v0(i), 5)
    w(i) = w(i) + delta_w(i)
    w(i) = Math.Round(w(i), 5)
Next i

i = 0
w0 = w0 + delta_w0
w0 = Math.Round(w0, 5)
Next j
kuadrat_Error_rerata = kuadrat_Error_rerata / max_epoh
kuadrat_Error_rerata = Math.Round(kuadrat_Error_rerata, 4)
kuadrat_Error_rerata = kuadrat_Error_rerata
If kuadrat_Error_rerata < target_Error Then
  Exit Do
End If
kuadrat_Error_rerata = kuadrat_Error_rerata - 
  (kuadrat_Error_rerata)
If kuadrat_Error_rerata < target_Error Then
  Exit Do
End If
Loop Until Epoh = Val(txtEpoch.Text)
pb.Value = 100
End Sub
End Class

2. Recognition MLVQ

Imports System.Data.OleDb
Imports System.IO
Imports Microsoft.VisualBasic
Imports System.IO.StreamReader
Imports System.Text

Public Class frmRecogKombi
  Dim Cari As Boolean
  Dim Tambah As Boolean
  Dim StrDB As String
  Dim ItemList1 As New ArrayList()
  Dim ItemList2 As New ArrayList()
  Dim ItemListN1 As New ArrayList()
  Dim ItemListN2 As New ArrayList()
  Dim str1(5) As String
  Dim BinerImg As String
  Dim NamaFileTrain As String
  Dim LebarImg, TinggiImg As Integer
  Dim mask(9) As Integer
  Dim ResizeImg As Image
Private Sub frmLogin_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Call KoneKsi()
    CMD = New OleDbCommand("Delete from Hasil", Conn)
    RD = CMD.ExecuteReader
    RD.Close()
    PictureBox1.Image = Nothing
    pb.Visible = False
    btnIden.Enabled = False
End Sub

Private Sub btnIden_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnIden.Click
    btnIden.Enabled = False
    Call ImgProses()
    Call ReadPixel()
    Call Recognizer()
End Sub

Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cancel.Click
    Me.Close()
End Sub

Sub ReadPixel()

    Dim tempbmp1 As New Bitmap(PictureBox1.Image)
    Dim Red1 As Integer, Green1 As Integer, Blue1 As Integer
    Dim X, Y As Integer, Gray1 As Integer
   ItemList1.Clear()

    With (tempbmp1)
        For X = 0 To .Height - 1
            Application.DoEvents()
            For Y = 0 To .Width - 1
                Red1 = (.GetPixel(Y, X).R)
                If Red1 < 0 Then
                    Red1 = 0
                End If
                If Red1 > 255 Then
                    Red1 = 255
                End If

                Green1 = (.GetPixel(Y, X).G)
                If Green1 < 0 Then
                    Green1 = 0
                End If

                Gray1 = (Red1 + Green1) / 2
               ItemList1.Add(Gray1)
            Next Y
        Next X
    End With

End Sub
If Green1 > 255 Then
    Green1 = 255
End If

Blue1 = (.GetPixel(Y, X).B)

If Blue1 < 0 Then
    Blue1 = 0
End If

If Blue1 > 255 Then
    Blue1 = 255
End If
Gray1 = (Red1 + Green1 + Blue1) / 3
Dim Tr As Integer
If Gray1 < 128 Then
    Tr = 0
Else
    Tr = 1
End If
BinerImg = BinerImg & Tr
ItemList1.Add(Tr)
lbTarget.Items.Add(Tr)
Tr = 0
Gray1 = 0
Red1 = 0
Green1 = 0
Blue1 = 0
Next
Next
End With
pb.Value = 50

End Sub
Sub Recognizer()
    Dim i As Integer
    Dim j As Integer
    Dim nilBiner2 As String
    Dim lenBiner2 As Integer
    Dim jumData As Integer
    Dim Piksel1 As Single
    Dim Piksel2 As Single
    Dim SJarak As Single
    Dim Distance1 As Single
    Dim TDistance1 As Single
    Dim TDistance2 As Single
    Dim Distance2 As Single
    Dim Normal1 As Single
    Dim Normal2 As Single

Dim Euclidean As Single
Dim Nama As String
Dim strPiksel1 As String

For i = 0 To ItemList1.Count - 1
  Application.DoEvents()
  Piksel1 = ItemList1.Item(i)
  strPiksel1 = strPiksel1 & Piksel1
  Distance1 = Distance1 + Piksel1 ^ 2
  Piksel1 = 0
Next

i = 0
TDistance1 = Math.Sqrt(Distance1)
For i = 0 To ItemList1.Count - 1
  Application.DoEvents()
  Piksel1 = ItemList1.Item(i)
  Normal1 = Piksel1 / TDistance1
  ItemListN1.Add(Normal1)
  Piksel1 = 0
  Normal1 = 0
Next

i = 0

CMD = New OleDbCommand("select * from TrainKombi", Conn)
RD = CMD.ExecuteReader
Do While RD.Read
  jumData = jumData + 1
Loop
RD.Close()
CMD.Dispose()

'Baca Biner database
CMD = New OleDbCommand("select * from TrainKombi order by ID asc", Conn)
RD = CMD.ExecuteReader
Do While RD.Read
  Application.DoEvents()
  Nama = ""
  nilBiner2 = RTrim(RD.Item("Biner"))
  lenBiner2 = Len(nilBiner2)
  Nama = RD.Item("NmFile")
  UserID = RD.Item("UserID")

  For j = 0 To lenBiner2 - 1
    Application.DoEvents()
    ItemList2.Add(Microsoft.VisualBasic.Mid(nilBiner2, j + 1, 1))
  Next
  j = 0
  pb.Value = 70
For i = 0 To ItemList2.Count - 1
   Application.DoEvents()
   Piksel2 = ItemList2.Item(i)
   Distance2 = Distance2 + Piksel2 ^ 2
   Piksel2 = 0
Next
i = 0
Piksel2 = 0
TDistance2 = Math.Sqrt(Distance2)

'Normalisasi citra db
For i = 0 To ItemList2.Count - 1
   Application.DoEvents()
   Piksel2 = ItemList2.Item(i)
   Normal2 = Piksel2 / TDistance2
   ItemListN2.Add(Normal2)
   Piksel2 = 0
   Normal2 = 0
Next
i = 0
Piksel2 = 0
Normal2 = 0

'Hitung jarak antar normalisasi
For i = 0 To ItemListN1.Count - 1
   Application.DoEvents()
   Normal1 = ItemListN1.Item(i)
   Normal2 = ItemListN2.Item(i)
   SJarak = SJarak + (Normal1 - Normal2) ^ 2
   Normal1 = 0
   Normal2 = 0
Next
Euclidean = Math.Sqrt(SJarak)
'MsgBox("" & Euclidean)
Dim Nilai As Single = Euclidean * 100
If Nilai = 0 Then Nilai = 110
CMD1 = New OleDbCommand("Insert into Hasil (Nama,Nilai,UserID) values (" & Nama & "," & Nilai & "," & UserID & ")", Conn)
RD1 = CMD1.ExecuteReader()
RD1.Close()
CMD1.Dispose()
ItemList2.Clear()
ItemListN2.Clear()
Loop
RD.Close()
CMD.Dispose()
ItemListN1.Clear()
ItemList2.Clear()
ItemListN2.Clear()
ItemList1.Clear()

'Cari nilai jarak terendah
Dim NamaNya As String
Dim NilaiNya As Single
Dim bolAda As Boolean
CMD = New OleDbCommand("Select * from Hasil order by Nilai asc", _
    Conn)
RD = CMD.ExecuteReader()
RD.Read()
If RD.HasRows = True Then
    NamaNya = RD.Item("Nama")
    NilaiNya = RD.Item("Nilai")
    bolAda = True
    UserID = RD.Item("UserID")
Else
    bolAda = False
End If
RD.Close()
CMD.Dispose()
If bolAda = True Then
    CMD = New OleDbCommand("Select * from tUser where UserID=" & _
        UserID & ",", Conn)
    RD = CMD.ExecuteReader()
    If RD.Read() Then
        Nama = RD.Item("UserName")
    Else
        Nama = "Tidak dikenal"
    End If
    lblNama.Text = Nama
    RD.Close()
    CMD.Dispose()
    CMD = New OleDbCommand("Delete from Hasil", Conn)
    RD = CMD.ExecuteReader
    RD.Close()
    CMD.Dispose()
Else
    Nama = "Tidak dikenal"
    lblNama.Text = Nama
End If

pb.Value = 100
ItemList1.Clear()
BinerImg = ""
salah:
If Err.Number <> 0 Then
Close()
End If
End Sub

Private Sub cmdLoad_Click(ByVal sender As System.Object, ByVal e As EventArgs) Handles cmdLoad.Click
cmdLoad.Enabled = False
OpenFileDialog1.Filter = "File Wajah|*.bmp;*.jpg"
OpenFileDialog1.Title = "File Wajah"
If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
PictureBox1.Image = Image.FromFile(OpenFileDialog1.FileName)
lblNmFile.Text = OpenFileDialog1.FileName
NamaFileTrain = OpenFileDialog1.FileName
btnIden.Enabled = True
End If
End Sub

Sub ImgProses()
pb.Value = 20
Dim LebarOri, TinggiOri As Integer
Dim bmp As New Bitmap(NamaFileTrain)
LebarOri = bmp.Width.ToString()
TinggiOri = bmp.Height.ToString()
If LebarOri > 100 And TinggiOri > 100 Then
LebarImg = 100
TinggiImg = 100
Rectangle.BackgroundImage = Image.FromFile(NamaFileTrain)
Dim newSize As New Size(100, 100)
ResizeImg = New Bitmap(Rectangle.BackgroundImage, newSize)
Rectangle.BackgroundImage = ResizeImg
Rectangle.BackgroundImageLayout = ImageLayout.Stretch
Else
Rectangle.BackgroundImage = Image.FromFile(NamaFileTrain)
End If

'Grayscale Proses
Dim source As New Bitmap(Rectangle.BackgroundImage)
Dim red, green, blue, grayscale As Integer
RectangleShape2.BackgroundImage = Nothing
For y As Integer = 0 To source.Height - 1
For x As Integer = 0 To source.Width - 1
red = source.GetPixel(x, y).R
green = source.GetPixel(x, y).G
blue = source.GetPixel(x, y).B
grayscale = red * 0.299 + green * 0.587 + blue * 0.114
source.SetPixel(x, y, Color.FromArgb(grayscale, grayscale, grayscale))

Next x
Next y
RectangleShape2.BackgroundImage = source
Dim img As Bitmap = New Bitmap(RectangleShape2.BackgroundImage)
Dim c As Color
Dim ii As Integer = 0
Do While (ii < img.Width)
    Dim jj As Integer = 0
    Do While (jj < img.Height)
        If (((ii - 1) >= 0) AndAlso ((jj - 1) >= 0)) Then
            c = img.GetPixel((ii - 1), (jj - 1))
            mask(0) = Convert.ToInt16(c.R)
        Else
            mask(0) = 0
        End If
        If (((jj - 1) >= 0) AndAlso (ii + 1 < img.Width)) Then
            c = img.GetPixel((ii + 1), (jj - 1))
            mask(1) = Convert.ToInt16(c.R)
        Else
            mask(1) = 0
        End If
        If ((jj - 1) >= 0) Then
            c = img.GetPixel(ii, (jj - 1))
            mask(2) = Convert.ToInt16(c.R)
        Else
            mask(2) = 0
        End If
        If (ii + 1 < img.Width) Then
            c = img.GetPixel((ii + 1), jj)
            mask(3) = Convert.ToInt16(c.R)
        Else
            mask(3) = 0
        End If
        If ((ii - 1) >= 0) Then
            c = img.GetPixel((ii - 1), jj)
            mask(4) = Convert.ToInt16(c.R)
        Else
            mask(4) = 0
        End If
        If (((ii - 1) >= 0) AndAlso (jj + 1 < img.Height)) Then
            c = img.GetPixel((ii - 1), (jj + 1))
            mask(5) = Convert.ToInt16(c.R)
        Else
        End If
    End Do
    ii = ii + 1
End Do
mask(5) = 0
End If

If (jj + 1 < img.Height) Then
    c = img.GetPixel(ii, (jj + 1))
    mask(6) = Convert.ToInt16(c.R)
Else
    mask(6) = 0
End If

If ((ii + 1 < img.Width) AndAlso (jj + 1 < img.Height)) Then
    c = img.GetPixel((ii + 1), (jj + 1))
    mask(7) = Convert.ToInt16(c.R)
Else
    mask(7) = 0
End If

c = img.GetPixel(ii, jj)
mask(8) = Convert.ToInt16(c.R)
Dim sum As Integer = 0
Dim u As Integer = 0
Do While (u < 9)
    sum = (sum + mask(u))
    u = (u + 1)
Loop
sum = (sum / 9)
img.SetPixel(ii, jj, Color.FromArgb(sum, sum, sum))
jj = (jj + 1)
Loop
ii = (ii + 1)
Loop
pb.Value = 30
RectangleShape3.BackgroundImage = img

Dim source1 As New Bitmap(RectangleShape3.BackgroundImage)
Dim sobelResult As New Bitmap(RectangleShape3.BackgroundImage)
Dim sobelX, sobelY, magnitude As Integer
Dim neighbourList As ArrayList = New ArrayList
For y As Integer = 0 To source1.Height - 1
    For x As Integer = 0 To source1.Width - 1
        neighbourList.Clear()
        neighbourList = getThreeNeighbourList(x, y, source1)
        sobelX = getSobelValue(neighbourList, "X")
        sobelY = getSobelValue(neighbourList, "Y")
        magnitude = Math.Sqrt(Math.Pow(sobelX, 2) + _
            Math.Pow(sobelY, 2))
        If magnitude > 255 Then
            magnitude = 255
        End If
    Next
Next

Universitas Sumatera Utara
End If

sobelResult.SetPixel(x, y, Color.FromArgb(magnitude, magnitude, magnitude))
Next x
Next y
picSobel.Image = sobelResult
picSobel.SizeMode = PictureBoxSizeMode.StretchImage
PictureBox1.Image = picSobel.Image
End Sub

Public Function getThreeNeighbourList(ByVal xPos As Integer, ByVal yPos As Integer, ByVal source As Bitmap) As ArrayList
Dim neighbourList As ArrayList = New ArrayList
Dim xStart, xFinish, yStart, yFinish As Integer
For y As Integer = yStart To yFinish
    For x As Integer = xStart To xFinish
        If (x < 0) Or (y < 0) Or (x > source.Width - 1) Or (y > source.Height - 1) Then
            neighbourList.Add(0)
        Else
            pixel = source.GetPixel(x, y).R
            neighbourList.Add(pixel)
        End If
    Next x
Next y
Return neighbourList
End Function

Public Function getSobelValue(ByVal neighbourList As ArrayList, ByVal maskType As String) As Integer
Dim result As Integer = 0
Dim sobelX As Integer() = {{-1, 0, 1}, {-2, 0, 2}, {-1, 0, 1}}
Dim sobelY As Integer() = {{1, 2, 1}, {0, 0, 0}, {-1, -2, -1}}
Dim count As Integer = 0

If (maskType.Equals("X")) Then
    For y As Integer = 0 To 2
        For x As Integer = 0 To 2
            result = result + (sobelX(x, y) * Convert.ToInt16(neighbourList(count)))
            count = count + 1
        Next x
    Next y
End If
ElseIf (maskType.Equals("Y")) Then
    For y As Integer = 0 To 2
        For x As Integer = 0 To 2
            result = result + (sobelY(x, y) * Convert.ToInt16(neighbourList(count)))
            count = count + 1
        Next x
        Next y
    ElseIf (maskType.Equals("X")) Then
        For x As Integer = 0 To 2
            For y As Integer = 0 To 2
                result = result + (sobelX(x, y) * Convert.ToInt16(neighbourList(count)))
                count = count + 1
            Next x
            Next y
        End If
    End If
    Return result
End Function

Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
    PictureBox1.Image = Nothing
    Rectangle.BackgroundImage = Nothing
    RectangleShape2.BackgroundImage = Nothing
    RectangleShape3.BackgroundImage = Nothing
    picSobel.Image = Nothing
    lblNama.Text = ""
    lblNmFile.Text = ""
    NamaFileTrain = ""
    cmdLoad.Enabled = True
    lbTarget.Items.Clear()
End Sub
End Class