

```

Sub CreatePresentation()
    Dim pptApp As Object
    Dim pptPres As Object
    Dim slideIndex As Integer

    ' Create a new instance of PowerPoint
    Set pptApp = CreateObject("PowerPoint.Application")
    pptApp.Visible = True

    ' Create a new presentation
    Set pptPres = pptApp.Presentations.Add

    ' Add title slide with specific layout and text
    slideIndex = pptPres.Slides.Count + 1
    pptPres.Slides.Add slideIndex, ppLayoutTitle
    With pptPres.Slides(slideIndex)
        .Shapes(1).TextFrame.TextRange.Text = "Plant Disease Detection Using
ResNet50"
        .Shapes(1).TextFrame.TextRange.Font.Name = "Arial"
        .Shapes(1).TextFrame.TextRange.Font.Size = 36 ' Change the font size as
required
        .Shapes(1).TextFrame.TextRange.Font.Bold = True
        .Shapes(1).TextFrame.TextRange.Font.Color.RGB = RGB(0, 51, 102) ' Change
the font color as required

        .Shapes(2).TextFrame.TextRange.Text = "An Experimental Study"
        .Shapes(2).TextFrame.TextRange.Font.Name = "Arial"
        .Shapes(2).TextFrame.TextRange.Font.Size = 20 ' Change the font size as
required
        .Shapes(2).TextFrame.TextRange.Font.Color.RGB = RGB(0, 51, 102) ' Change
the font color as required
    End With

    ' Loop through each section of the presentation and add slides with content
and images
    AddSlideWithCustomization pptPres, "ResNet50", "ResNet50 is a deep
convolutional neural network architecture introduced by Microsoft Research. It
consists of 50 layers and utilizes residual connections, allowing it to
effectively train very deep networks. ResNet50 has achieved state-of-the-art
performance in various computer vision tasks, including image classification and
object detection. Its architecture comprises multiple building blocks, including
convolutional layers, batch normalization, and global average pooling. ResNet50
is widely used in practice due to its excellent performance and ease of
training. It has been pretrained on large datasets such as ImageNet, making it
suitable for transfer learning in various applications.", , 32, 24
    AddSlideWithCustomization pptPres, "Dataset Description", "The dataset used
in this study comprises thousands of high-resolution images of various plant
diseases affecting different crop species. Each image is labeled with the
corresponding disease type and severity level. The dataset is meticulously
curated to cover a diverse range of crops, diseases, and environmental
conditions, ensuring robustness and generalizability. Additionally, the dataset
includes healthy plant images to serve as negative examples during model
training and evaluation.", , 32, 24

```

AddSlideWithCustomization pptPres, "ResNet50 Architecture", "ResNet50 is a deep convolutional neural network architecture consisting of 50 layers. It introduces the concept of residual connections, allowing for the training of very deep networks without suffering from the vanishing gradient problem. ", "C:\Users\abhay\Downloads\resnet50.jpg", 32, 24

AddSlideWithCustomization pptPres, "ResNet50 Architecture", "The architecture comprises multiple residual blocks, each containing convolutional layers, batch normalization, and ReLU activation functions. ResNet50 employs skip connections to propagate information across layers, facilitating smoother gradient flow and better feature learning. With its superior performance and efficient training process, ResNet50 has become a popular choice for various computer vision tasks, including image classification, object detection, and semantic segmentation.", , 32, 24

AddSlideWithCustomization pptPres, "Training Process", "The training process for the ResNet50 model involves several key steps. First, the dataset is preprocessed to normalize pixel values and augment the data for increased robustness. Next, the model is initialized with pre-trained weights on large-scale datasets such as ImageNet. During training, the model learns to minimize a loss function (e.g., cross-entropy) by adjusting its parameters through backpropagation and gradient descent optimization. Regularization techniques such as dropout and weight decay may be applied to prevent overfitting. The training process typically spans multiple epochs, with the model's performance monitored on a separate validation set to prevent overfitting and guide hyperparameter tuning. Finally, once training is complete, the trained model can be evaluated on a held-out test set to assess its generalization performance.", , 32, 24

AddSlideWithCustomization pptPres, "Evaluation and Results", "The ResNet50 model achieved high accuracy and performance in detecting plant diseases, as evidenced by evaluation metrics such as accuracy, precision, recall, and F1-score. The results demonstrate the effectiveness of ResNet50 for plant disease detection tasks.", "C:\Users\abhay\OneDrive\Pictures\Screenshots\Screenshot 2024-05-04 230402.png", 32, 24

AddSlideWithCustomization pptPres, "Conclusion", "In conclusion, the presentation highlighted the effectiveness of the ResNet50 architecture for plant disease detection, leveraging deep learning techniques. The study demonstrated the model's ability to accurately classify and detect various plant diseases across different crop species. By automating the detection process, ResNet50 offers significant potential for improving crop yields and ensuring food security. However, further research is warranted to enhance model robustness and scalability for real-world deployment.", , 32, 24

```
' Add the last slide with a "Thank You" message
slideIndex = pptPres.Slides.Count + 1
pptPres.Slides.Add slideIndex, ppLayoutTitle
With pptPres.Slides(slideIndex)
    .Shapes(1).TextFrame.TextRange.Text = "Thank You"
    .Shapes(1).TextFrame.TextRange.Font.Name = "Arial"
    .Shapes(1).TextFrame.TextRange.Font.Size = 77 ' Change the font size as
required
    .Shapes(1).TextFrame.TextRange.Font.Bold = True
    .Shapes(1).TextFrame.TextRange.Font.Color.RGB = RGB(0, 51, 102) ' Change
the font color as required
End With
```

```

' Clean up
Set pptPres = Nothing
Set pptApp = Nothing
End Sub

Sub AddSlideWithCustomization(pptPres As Object, title As String, content As
String, Optional imagePath As String = "", Optional titleSize As Integer = 32,
Optional contentSize As Integer = 20)
    Dim slideIndex As Integer
    Dim slide As Object
    Dim pic As Object

    ' Determine the index for the new slide
    slideIndex = pptPres.Slides.Count + 1

    ' Add a slide with text layout
    Set slide = pptPres.Slides.Add(slideIndex, ppLayoutText)

    ' Set title and content
    With slide
        ' Title
        .Shapes(1).TextFrame.TextRange.Text = title

        ' Content
        .Shapes(2).TextFrame.TextRange.Text = content

        ' Image
        If imagePath <> "" Then
            Set pic = .Shapes.AddPicture(imagePath, False, True, 100, 100, 500,
300)

            ' Resize the image to cover a good portion of the slide
            With pic
                .LockAspectRatio = msoTrue ' Maintain aspect ratio
                .Width = .Width * 0.9 ' Resize width to 80% of slide width
                .Height = .Height * 0.9 ' Resize height to 80% of slide height
                .Left = (pptPres.PageSetup.SlideWidth - .Width) / 2 ' Center
horizontally
                .Top = (pptPres.PageSetup.SlideHeight - .Height) - 0.5 / 2 '
Center vertically
            End With
        End If

        ' Apply styles
        ' Title font style
        With .Shapes(1).TextFrame.TextRange.Font
            .Name = "Calibri (Body)"
            .Size = titleSize
            .Bold = True
            .Color.RGB = RGB(0, 51, 102) ' Dark blue color
        End With

        ' Content font style

```

```
With .Shapes(2).TextFrame.TextRange.Font
    .Name = "Calibri (Body)"
    .Size = contentSize
    .Color.RGB = RGB(51, 102, 153) ' Light blue color
End With

' Apply background color
.FollowMasterBackground = msoFalse
.Background.Fill.BackColor.RGB = RGB(255, 255, 255) ' White color
End With

' Clean up
Set pic = Nothing
Set slide = Nothing
End Sub
```