Title - Face Mask Detection

Group Members -

2127012 - Rohit Dhotre

2127014 - Akash Gadade

2127017 - Shubham Gaikwad

2127004 - Sneha Bhaskar

2127033 - Harshada Khuspe

TOPIC

Progress Report -II



```
In [1]:
            import cv2
              img = cv2.imread('E:\img_1.png')
 In [3]:
              img.shape
 Out[3]: (675, 1200, 3)
 In [4]:
              img[0]
 Out[4]: array([[157, 141, 129],
[157, 141, 129],
[158, 142, 130],
                        ...,
[125, 132, 129],
[125, 132, 129],
[124, 131, 128]], dtype=uint8)
 In [5]: import matplotlib.pyplot as plt
 In [6]: plt.imshow(img)
 Out[6]: <matplotlib.image.AxesImage at 0x2c03b1514c0>
               0
              100
              200
              300
              400
              500
              600
                                                600
                           200
                                                                    1000
                                                          800
             while True:
    cv2.imshow('result' ,img)
    if cv2.waitKey(2) == 27:
    #27 is the ASCII of Escape
              break
cv2.destroyAllWindows()
 In [8]:
              haar\_data = cv2. Cascade Classifier('C: \Anaconda \Lib \site-packages \cv2 \data \haarcascade\_front alface\_default.xml')
 In [9]:
              haar_data.detectMultiScale(img)
                                   89, 264, 264],
26, 463, 463],
115, 250, 250],
584, 56, 56]])
 Out[9]: array([[ 198,
In [10]: # cv2.rectangle(img, (x,y), (w, h), (b,g,r), border\_thickness)
              while True:
    faces = haar_data.detectMultiScale(img)
    for x,y,w,h in faces:
        cv2.rectangle(img, (x,y),(x+w, y+h), (255,0,255), 4)
    cv2.imshow('result' ,img)
    if cv2.waitKey(2) == 27:
    #27 is the ASCII of Escape
        break
              #27 is the ASCII of
break
cv2.destroyAllWindows()
In [23]:
              0
             1
2
3
4
5
6
7
8
9
10
11
12
             13
14
             15
             16
17
18
19
20
21
22
23
```

23		
24 25 26 27 28 29		
27 28		
29 30		
30 31 32 33		
33 34		
34 35 36		
37 38		
39 40		
41 42		
43 44		
45 46		
47		
48 49 50		
51		
52 53		
53 54 55 56		
57		
58 59		
60 61		
62 63		
64 65		
66 67		
68 69		
70 71		
72		
74 75 76 77		
76 77		
78 79 80		
80 81		
82 83		
84 85		
86 87		
88		
89 90		
91 92		
93 94		
94 95 96		
94 95 96 97 98		
94 95 96 97 98 99		
94 95 96 97 98 99 100 101		
94 95 96 97 98 99 100 101 102		
94 95 96 97 98 99 100 101 102 103 104		
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 111 112 113		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 111 111 112 113 114 115 116 117 118 119 120		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 111 112 113 114 115 116 117 118 119 119 111 1115 116 117 118 119 119 110 111 111 111 111 111		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 111 112 113 114 115 116 116 117 118 119 120 121 121 122 123 124		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 111 112 113 114 115 116 117 118 119 120 121 121 122 123 124 125 126		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 111 112 113 114 115 116 117 118 119 120 121 121 122 123 124 125 126		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 111 111 111 111 111		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 111 112 113 114 115 116 117 118 119 120 121 121 122 123 124 125 126 127 128 129 130 131 132 133 134		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 1112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 131 131 131 131 131 131 131 13		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 1112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 131 131 131 131 131 131 131 13		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 1112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 131 131 131 131 131 131 131 13		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 137 138 137 138 137 138 139 140 140 140 140 140 140 140 140		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 131 138 139 140 131 132 133 134 135 136 137 138 139 130 131 131 132 133 134 135 136 137 138 139 130 131 131 132 133 134 135 136 137 138 139 130 131 131 132 133 134 135 136 137 138 139 130 131 131 132 133 134 135 136 137 138 139 130 131 131 132 133 134 135 136 137 138 139 130 131 131 132 133 134 135 136 137 138 139 139 130 131 132 133 134 135 136 137 138 139 139 130 131 132 133 134 135 136 137 138 139 139 130 131 132 133 134 135 136 137 138 139 139 139 139 139 140 140 140 140 140 140 140 140		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 131 132 133 134 135 136 137 138 139 140 141 140 141 140 141 141 151 161 171 181 191 191 191 191 191 191 19		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 140 141 141 151 161 177 188 199 190 191 191 191 191 191 191		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 141 141 151 161 177 188 199 190 191 191 191 191 191 191		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 140 141 140 141 140 141 141		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 141 141 141 151 161 177 188 199 190 191 191 191 191 191 191		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 141 141 141 151 161 177 188 199 190 191 191 191 191 191 191		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 141 141 155 166 177 188 199 190 191 191 191 191 191 191		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 133 134 135 136 137 138 139 140 141 140 141 141 155 166 177 188 199 190 191 191 191 191 191 191		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 140 141 141 151 161 177 188 199 190 191 191 191 191 191 191		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 140 141 141 142 143 144 145 146 147 148 149 150 151 150 151 150 151 151 151		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 1112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 141 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 150 151 152 153 154 155 156 157 158 159 150 151 152 153 154 155 156 157 158 159 159 159 150 151 152 153 154 155 156 157 158 159 159 150 151 152 153 154 155 156 157 158 159 159 150 151 151 152 153 154 155 156 157 158 159 159 150 151 151 152 153 154 155 156 157 158 159 159 150 151 151 152 153 154 155 156 157 158 159 159 159 150 151 151 152 153 154 155 156 157 158 159 159 150 151 152 153 154 155 156 157 158 159 159 159 159 159 159 159 159		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 1112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 131 133 134 135 136 137 138 139 140 141 141 151 161 161 163 164 165 166 167 168 169		1/1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166		1/1

```
173
174
175
176
177
178
179
180
181
182
183
184
185
         186
187
188
189
190
191
192
193
         194
195
         196
197
198
199
In [13]: import numpy as np
In [14]: x = np.array([3,2,54,6])
In [15]:
Out[15]: array([ 3, 2, 54, 6])
In [16]: x[0:2]
Out[16]: array([3, 2])
In [17]: x = \text{np.array}([[3,4,54,67,8,8],[1,2,2,4,5,7],[4,5,3,5,6,7],[1,2,3,34,6,8]])
In [18]: x
In [19]: x[0][1:4]
Out[19]: array([ 4, 54, 67])
In [20]: x[0:3,0:3]
In [21]: x[:,1:4]
In [ ]:
In [22]: np.save('without_mask.npy',data)
In [24]: np.save('with_mask.npy',data)
In [25]: plt.imshow(data[0])
Out[25]: <matplotlib.image.AxesImage at 0x2c04b8bb7f0>
         10
 In [ ]:
```



Face Mask Detection

