

```
{ "cells": [ { "cell_type": "code", "execution_count": null, "metadata": { "id":
"hrCz_igRIDEz" }, "outputs": [], "source": [ "!pip install -q kaggle" ] }, { "cell_type":
"code", "execution_count": null, "metadata": { "colab": { "base_uri":
"https://localhost:8080/", "height": 90 }, "id": "f2OEtAk0ISIn", "outputId": "b1099878-d763-
47c9-da8b-e47c7c0fd219" }, "outputs": [ { "data": { "text/html": [ "\n", "\n", "\n", " Upload
widget is only available when the cell has been executed in the\n", " current browser session.
Please rerun this cell to enable.\n", "\n", " " ], "text/plain": [ "" ] }, "metadata": { },
"output_type": "display_data" }, { "name": "stdout", "output_type": "stream", "text": [
"Saving kaggle.json to kaggle.json\n" ] }, { "data": { "text/plain": [ "{ 'kaggle.json':
b'{'\"username\": \"samyuktharenu\", \"key\": \"c208fb742e7909d1ce586ebdbfc06a7\"}'" ] },
"execution_count": 2, "metadata": { }, "output_type": "execute_result" } ], "source": [ "from
google.colab import files\n", "files.upload()" ] }, { "cell_type": "code", "execution_count":
null, "metadata": { "id": "xEeUuLSrIdq4" }, "outputs": [], "source": [ "! mkdir ~/.kaggle" ] },
{ "cell_type": "code", "execution_count": null, "metadata": { "id": "U1PiwZw5ImpZ" },
"outputs": [], "source": [ "! cp kaggle.json ~/.kaggle/" ] }, { "cell_type": "code",
"execution_count": null, "metadata": { "id": "BF0oq57rIvzz" }, "outputs": [], "source": [ "!
chmod 600 ~/.kaggle/kaggle.json" ] }, { "cell_type": "code", "execution_count": null,
"metadata": { "colab": { "base_uri": "https://localhost:8080/" }, "id": "BXgfMJmQIzov",
"outputId": "6791bd1a-e7b6-4acf-c8dd-0b2a7a9fb51d" }, "outputs": [ { "name": "stdout",
"output_type": "stream", "text": [ "Downloading animal-image-dataset-90-different-
animals.zip to /content\n", "100% 656M/656M [00:16:2: UserWarning: `Model.fit_generator`
is deprecated and will be removed in a future version. Please use `Model.fit`, which supports
generators.\n", "
model.fit_generator(xtrain,steps_per_epoch=len(xtrain),epochs=10,validation_data=xtest,valida
tion_steps=len(xtest))\n" ] }, { "name": "stdout", "output_type": "stream", "text": [ "Epoch
1/10\n", "54/54 [=====] - 142s 3s/step - loss: 4.8618 -
accuracy: 0.0198 - val_loss: 4.3057 - val_accuracy: 0.0398\n", "Epoch 2/10\n", "54/54
[=====] - 137s 3s/step - loss: 4.2082 - accuracy: 0.0557 -
val_loss: 4.0744 - val_accuracy: 0.0713\n", "Epoch 3/10\n", "54/54
[=====] - 137s 3s/step - loss: 3.8943 - accuracy: 0.1050 -
val_loss: 3.7169 - val_accuracy: 0.1370\n", "Epoch 4/10\n", "54/54
[=====] - 136s 3s/step - loss: 3.5995 - accuracy: 0.1433 -
val_loss: 3.3429 - val_accuracy: 0.2000\n", "Epoch 5/10\n", "54/54
[=====] - 136s 3s/step - loss: 3.3404 - accuracy: 0.2028 -
val_loss: 3.0638 - val_accuracy: 0.2524\n", "Epoch 6/10\n", "54/54
[=====] - 136s 3s/step - loss: 3.1139 - accuracy: 0.2489 -
val_loss: 2.9100 - val_accuracy: 0.2872\n", "Epoch 7/10\n", "54/54
[=====] - 153s 3s/step - loss: 2.8753 - accuracy: 0.2930 -
val_loss: 2.5939 - val_accuracy: 0.3517\n", "Epoch 8/10\n", "54/54
[=====] - 135s 3s/step - loss: 2.6798 - accuracy: 0.3298 -
val_loss: 2.4989 - val_accuracy: 0.3728\n", "Epoch 9/10\n", "54/54
[=====] - 135s 3s/step - loss: 2.5094 - accuracy: 0.3689 -
val_loss: 2.3658 - val_accuracy: 0.3985\n", "Epoch 10/10\n", "54/54
[=====] - 135s 3s/step - loss: 2.3363 - accuracy: 0.4107 -
val_loss: 2.2267 - val_accuracy: 0.4320\n" ] }, { "data": { "text/plain": [ "" ] },
"execution_count": 23, "metadata": { }, "output_type": "execute_result" } ], "source": [ "#
train the model\n",
"model.fit_generator(xtrain,steps_per_epoch=len(xtrain),epochs=10,validation_data=xtest,va
lidation_steps=len(xtest))\n" ] }, { "cell_type": "code", "execution_count": null, "metadata":
{ "id": "qk20RhGBQGHH" }, "outputs": [], "source": [ "# save model\n", "\n",
```

```
"model.save('animal3.h5')" ] }, { "cell_type": "code", "execution_count": null, "metadata": {
"id": "EtoY6OW1Tntj" }, "outputs": [], "source": [ "## Testing model" ] }, { "cell_type":
"code", "execution_count": null, "metadata": { "id": "EEYjECGuT640" }, "outputs": [],
"source": [ "from tensorflow.keras.preprocessing import image\n", "import numpy as np" ] },
{ "cell_type": "code", "execution_count": null, "metadata": { "id": "EB9SV9qlUDgd" },
"outputs": [], "source": [ "## Testing 1\n", "\n", "img=
image.load_img(r'/content/animals/animals/butterfly/16a79172c2.jpg') #read image\n", "\n", "
" ] }, { "cell_type": "code", "execution_count": null, "metadata": { "colab": { "base_uri":
"https://localhost:8080/", "height": 81 }, "id": "_H1dFCHfiC_s", "outputId": "65ba1bd8-fce5-
449f-a7bc-62e59030c345" }, "outputs": [ { "data": { "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAEAAAABACAIAAAAlC+aJAAAIVUUEQVR4nFV
6Z7xnRZF2VXX3Cf9088ydmTs5DxIJggurKLogQYIgScAFEVd0VUB3EVSCMoAEFQRm
USQICCPJUFFAMMCQhgGGgQlMnrn5/tM5p0PV++Hq++77fDofqv361PVVc9TXfi9i/c
WTV48CRhBIGH2FgWFvfdBwDnJrXTEyYCDKaZsVKSUCYV1Rd5w+aaieGu03malkZR
SEZISFFXj1CRUNlprXYriUpIahSlIYHc2Rly7LYQkEIKa0hpMEleIjePgnAOlAdiF3AUvg
0pShZodFuWaTINsZKMtz64Z5YVr+3o9H9XBAynlbKEQESgCUIKIYCKEijWGUGhDaz
oaSKu9GCMopYyXom15gttZ5pjZ2hAAkyTx7AtNMXvF2jIDAAIRKaUQRAXGvqRze7v
NzAhIRJEyaVyOTGowCULWWiYBYJfKnhcxVZOUHOXAxbhtyERxXG7zSBTHyJGDU
mQKzUGDD2LFKIDEjARECEGhEhFmVh66KU7jpKKkpFNyDlgsC06Cy6XhvGcVOLAA
F3kliUVMCCF4FQgjNqJJUECUBIBoCJNyUWO0PYJBR5GOosgYk6hIUQxEChQjAwZBji
hVWDIRAAKA2OACS6RjLVFGTpmqYqdNrJ0FCWItkHKsiVFrAABxwiGEGBfdd8+Tql
K97JROq0hMAC1Zo5EVfsjbHeDrAo6FSWlMRGgVuSAbJcikwfBk5IgAQAAJHEAo0Wm
EKaBHREURYUQURVEkiCDkAmrDoJgNgGgEnegOoxK0jOBAPBp3OD8m6GKICtBEHs
SCYtRMICjIwIGAmMEJTB93C6ZONDxZ/d1fbvnVy9tHgRwVzWoF04iSyJuSj0EIIeZBG0
EUQc6tb3ERggCw95aZOYhzzofAIg4FRFdMGUQRGhBDAQQRyQBpUEQKiCJCQ1qj8q
wiiuI4KpXSDvCBBdK0rLVGyg0roZTQs2IwSiIiJMfzKwZlIKjQ2Mjreyss4741QPfOOTD+
19816tRCtO6bLlbVWrUoV2qTKyoFBltiMUDADO32+12aObBZcG1bdF2RcE+hBBCAAC
IFBFFUQQAACAiSimtNREohUSAKP+4N4hoBAijJ17KNZNWVBD0VEtrQQWOvWKmR
KWoosjoxOg0jgyQJiIAEkAP9z7zsjIawuaXfvfcTXc8tU83XH3f+HC1W4ITaaIvtVJiIlCagT
xpdixWnGUZbUwUwdvg2962XGGdy8V6xYwBOCcCrTURKc1kNGPw4JAYiTUBiQYA
RCAlmoJoCkhRrONKrRBhJUKuk1JAGYjSkdKkgDQrjQoBgAEAEQFk2rRp+y9rXLvirm
OOPynkaVknG4fHe8RuX7N4xkAgx3EIOvhJe2MMAARh74GZnZdtgztn9E5NMQUgg5iS
RkTCyf8OWutJd4UQRARRRIJIYUgHkWEBQAECgAUQKVUmqYuTwGsUqpWKO/6uk
6QIIIGCABIGOD/Q7PZfGmw86F77xoa58V7zP3YRz80mMWYzr3spy9Ry8c2dAeqgPx/a1
CUUgAAQIW1mSty760LjsM/4f6v7T8tJ8GARDRGJErC/9owADJgCCEopSodNaWUUpR
RIDEeLJgkdgQIgoAV3XHnM5lMczNin7xRuHOO+95qqr4MOW/uYPL3zn8vNfGll3x/LT
T7h2CIYa5fFsD0kHGCJFSiNq1BpQCragYiE1PD7WzOqZa7WLvC3eSd4OhUXHFDw4
Dw4MCnofmMUhAiKIRjAIGACY2RMicisPmePcgzOxTqqRkoVatVwpp8KZRi2AgRQDI
IC88Nd1W7e9k0S02+z+kUHQ9fL9T1536YWXrLjl0all9YF9Fi+dRjN2n3XamUsfX7n2Y8l
QV6m6m040uS0gigiqOLEIigIggEEIYGhuNerUCYLLBYEQJE4QQBEEYGHIWRdpoB8YY
RARgRCDClEwKWLzPHYu2KKQVgouSxEszTqDDdw9PDGkvVinQwoCydOEhOLrGtjZ
945ILp+82o1Xkl550/hNXXfHKJnvwb+d8+/KLvtx+1/jO0OKie92On75dHLG4HNt2dwJzK
sl4rMaATTBeWAANCZEAgEgYb49SqYqFcUFrcrHSiKhBtNYMWmuV2UxrQmFFCkGR
Cqg0ewYAZsfBhsIGBsColAajoKIjoNaMKTw+XiFG77lw6ByrJS+sf+SJv9vW0MY3Vz/zs0
e/cMx/7Lv3gjkLFi8/btnaP60aHd96xpknSrl809dX3P3a1kd+dfF9jWZUHLswPaQ7AkuERP9
IxD/oBBEJnCe+yxzdjKlZkVuvSuCttxYrAs5g89s4dkzMJIgCZJVGPekBAfehSIPWeHaee
YtgDcRxmllk/bMWaI9ATKDFyv1J//4/rcu//rsPfou/fJNS/oqV6z4wc0Xf6W33P3LZ1fXuuY8
dd2Pb1011gnq/W27Bmrmyu/fpKbHODahVFLSUZf4HaIdMSrRyCLACJPBICHJbAEKIKh
SiNqaFbBG0OhNgIgrVRbAjupKUKjAYSQBANDsN62fR5C0KhIEGMSUI9WmEQJxhO9
2IIA12AsHIBUzzv1o7P6p73x+KvHfHDBX9ftzAd3DOFR0V+9GEPxP49cddZ3Lrtwv17NZ
```

mvW+tHtF3/9ks//baX92htOWMdRCVnSgPRPICIIgmAQ8CzeoS2Cd5J755ybTJ3ee8uhGdp
tnvRPlvvCBhsgCARGduxYvPfWuaJd5BONRI73thDvAgCoqNRVdjQIxyAeQoJ03V2P9/36
6XOP+tz9zz5w0SmfvPmy7+2z2z7ZltFli2YUbZtM7/7di0PLT1+sP3LyNdfe/peVG8vlyg7X2
0ianiAAekUsBpEBQIAYGAEEBAAsi2/7chQgjkLQgk6BJgzMCoAYcoOY5YSqSQoMAk
HsxNpQ2JB7zDz64NBZiI3KJmJXE8CgKVHIWDNCAAHWlkYP33Ph2m0T+5+88K5nZq
7dtOuq//mfdIYeXb+IXBT3X7b8mY3blnaXFlzwheKBm1av3VFYmNGF79ZHVzf753MoSJ
FOMOTyDwAiTNI4EREWB4o8AAZGhSgKwSiCEIJE+yCkGJm8a1uIEAMVXmzhCjtZ3d
izCKAAh6+rqLM1Ue6sBYi9wgoBCJJYZFV0ffpz/zZrSvWNP609cr/Fv391rW+Mguh3Vq/b
wDh3zrQTPnHIR8445v2n3xjpOOa7H/5AOVXvDo0dOGPqndvMKK12FLPWGIApItKIygc
MTAikQF60BM6s5I6s83nhrPPWhRCC9c5650JehKLebjTyVrvdbuaNdt4oXNtyK6ANIB4Z
wWe2PTaejQ+3Wi0RDyJKi5AwioQi4jsfeGDrSPXZ6J7Ldp/T7df+6eG7vrpyYPPUt33oJ7uy
71541G4nfPL911f+8L/vWLLurNaZthIcHq+PNZrs5too0Wi01sqrEAI/JOSIHOQwJM+sZaV
0d6xJQZANMTToAECk0MiA3HYISBEqkcDCQXwILEAKsVCAqMfHR3s6Zg0PUbWckU
HtPAsKAGj2XbNqx+x7ZF7P9zj2oF9/60/TTaON5oSTT85ye/Utv3h3rH3bEWeJidI4/8Z5x9
x2y/2PP/mjkPI11//s+t+t+dRxizxphYRKOQoakFmLCLMD0ASJhwwAixAAIMaQKwTEIKI
IAB0qQm1RTNu1EcWyUoqIJXhPAAaQCFJQIThmFh/rtsjM3pkks5THL527CBEVAAKglY
/M+eBDj/2FCuRmfbxcHx6EKnXGkZ3e3/X39TvKFKGy3eXOhbP6tw9tW9Dbfdy3zrzvvhc
2b6svnjI8Qty2rp711loRCEzeewBmJpcLkIgICiBiGmFkVKxQRayJkBXqMUaZiJWWJBKtI0
RUPMmSPIgJhVNM8gK9j5RSvd1T+/rN/CWj2okQ+cAaOQDC3ff+qVZKjjjhI+c+anPHX
5BUslvffjm3v55ZxxyxG9XPpw3x4c3rv3Wxbesfnv9icd/6p1tb5/9mVvHW9vbEnWfON/JIEt
FKwwEAKRFic8hURtXt2cvLjkrIJpFJNiCioXoAgckr0BppR2AlgqanjTvVh5jPVhAfOJCrV
AJBABCBAxkJaDEY6PjzLUoqRIH4kCBxbE4lptv/fauCZ/XfTVipUf2XrromYd+v+HFV5c
tHXjxlrt7dGlg6d7StA+sevjUb5z5Hxed/41/W1bp6dlz8cy7f7ZO8dz/TTMR1REHfgRDdcFu
CQBog4CTpFScC7nnzLG1wTtxlkOQKUbv0cmHdKZzKmqqgikxsZcQhAOA/KPAhxAQG
cCHEEZHGOpbWWsyVhTx5BFRrYgJN9x38/vfRJ25DSjk2bJ9aNDTf+7dzPPnHdPaOV
313zk8cW68Yvv/vAwJx5P7zlhgLiGdqlBhr/fvHRxegGDiCkQXsRzOKRJ1Y/QID1HFAiAE
DySMisfUCyQSsJuWED2qBRM13zDKV6QwQ6HUHx1N6uIPNAiESEgghBIIhXAUQwiPi
RYcRLvnRQPWuJ8GTCXvPoUKnbf02rFwwPDv3w9nsTo2++/8o5XYtt3ISIjt/32HK1khX
x4t7aurHtwyPNJx+643uXLn9y9YZWUj3s8AU6HpTg2IfIW5uVJjblbhY7aKIvAYD33jsMH
icVTMkQMJJGbaRWwkP7cUknTDFpVE6HOQzb9kujru1URGKMCY69w2bTuyIKnsgQE
ZFC6qx0dlIU6iRQiKaWvOO+EuNK5z8F7LNqtigkR0W3f+/mvrn3w3uV3nH30+RGnfWnp
qH0HSrVpV3zutFoanXLafy2bFqdi50/b/YmnXmT2yHr/vu7zBmao8dHWYgoexE4RkUIBP
Cm4vfeZC20bLEs756xg1/LYVGEicO7JgflcvGoECZ4mFRuhIYwAtPcCoEUEEUWEqmm
pr9pdUqWYEilm9gH9mGd33XLPbT/49Rkf+hdt/bb31t3/zEMnnX3uz399x0MvPvDpMz69
RXa2wtYHHI358D0rpg00aPbiBVN6f3zD8VgAulQEDk6S+VN7P3voHIWrfSFCEwVAUB
isZxfAM0qg3JF1EDx4B+QFXe5bOWSenDNW7W16kItzznntHSshcQEYCYAwkJAGFAYq
IUqVSmV631QShUxnnX/j+ER901ubJjL56XPPTgRZfusNdz39i1J/JzsdIN7vwE+M7Yqh3T
7znE9eff2Np51wzOsvr5JK58kfv/iC4z950ryhKUZUHHXO2nduR181tZPXOOtAZMwMYp
hBBDmAsyF4CR6dZSlo1wi7uo05nmjlg+18KKemw9z64CH4fyYb/w+ISURO2AJTrHSi4m
oUVZJSopJFs2cbYy77+dUTrebsGQOA3DV3qgTnfLuxZfycg44757Tj7YbGWxtGbrv/14cfu
s99P3zsqPNO+8a/f35gevGTX/5+Wb7ojBnld+r11VvW4PCOPbXa11BNFTEjM3hHIThh4o
AsxELoS0ARQV1Asy3DTdnZrO9st3YFjhClCByUc6EoXCvzLiD7f8plRmYwAdo2kShlO
JrdSWt3bLvy6nPvv+eZHW9vTLX2u8aXzZpy0bHnbduytTJl6kGH73X73+7TSkIrYqpffMr
Xlv/oF9d/56sXf/Wy/qk9OydUR2dlz5tffO1re9f6ajK+LYyPnpFTedbuVKk+tOu1298d9Q48i
wgyg4CISMGYEAsJBUIKwjpuSMhJBQUA2jtScpwr2KYavbBGRGEBEAEnQiCiiShSkW
WBcennu9On3/OLBJ59/608vvgAA60fy9tDgh/Y78KozP5YLSVG8u+LWFthn39512ufOeWb
tzu995rDDP//dX1517gePOPreG3+0cu2b/1Ov2bHmXvsf8M4OmKH/Xs1sgj2hp+80b9fWnn
92MLeQBmYWEVEhBOZAoI2WmLimVVcFumppUZA2krwVmL2wFm0gWPIaMSjUlizC
IkIqBoJJ1UcmAmNMHMDP/X29LXiwd7fkS2YCUfsu8fpHz94fGTiM8sffvDJlZ9+1eMzD
97zjIs+9v6zTwB2x99+jufO+I/fnTP5jfeiOfN2rNzeomg6DnwqOve+PPG6WrvpW5GedeGl

bqdb+ma/e7z9YFKITwzEzNxAGEMgt4hiIpiKXWqpItKXWmpo29qLW60bcFkrS8ysDkW
OXuHk50Z7733PnDBbEmQAzEJiNKRliOP2u+U078yZ3rvFz938sp1dr9le1j2V971217Do9t
3XPTRhcu/91DjXcgq0zRl3WRuuPsPt/37mV/+goq5LGnX15x+RfHFgxsXfn3l35+7XDXPr
es7/vBZj/89uPz6hvuPn3/AwKY8gxBDKy9cAARgYI9k4s6SZd1qUZRNYpKIagiLWl24Bza
Qmwh3qH33jMzAAsWgV0IRQAN6JiBNREHrcqLLqs3vH34h4/+7g+Xe6XKHYkSXLN11
Oloz0XzbMcBPeVNI0OW24aITjrlpK9G7a/dfOfu82ftnUbLp3RtWv33/37qvScv/8SM/r1Gh5
867tRa/rcFvMu2R4fSEh07kDyyflucGETwXoUQiJCZETGp6VDxUmPoZi9Wj/ueUkJEHRO
FCCwSAABFBCkgoIB471mAkHRAz+CFhMlprkybN3fNa39eOnvWOR85aO2GN5575pkv
fvwg0LbdN++2h39+3FG79XZQ/1zqUuaxxx8JpTQu0TNRNIE1XlqLH3xudbvVrE05pHrYJ9
YNbdkmuzZ2y1AYs/mExFQuRVtUjOM0iimOTRybVHcnYFSIfRRU6qksUsJAXplSJdWh
0AQ6sOdAntkG5ywLKxFRYERQCZMxJooiVijlXITVhrGLbr7l4IN2P/24lzZt39K35aUvL6
xesmRgxfmnfubDM5+46cJX/vxGaLswYsetvLJlIs/zS88511q/3VW65iy8/cvHz6wV723YIHb
6LTtkV3M4VOtmUY90GJ9wo6uDKULEKIqSVJcrsVSbr143Xp2CVMiW+QYWDoZVU
WgjCWFEELwDm0hbRss0yQRZAZSXmLQWgJBs1FKqQDRf11y8/fv/+MilUKtc9FBB7/
y0vo5i/fWVE6WLYkP2q9n/qKJvpkr393W6psls+eefRBpalTQlQudyftxLCJx3ass9RRU0n/
PvOjrvl5PWmPDJbLO9KZUW3OPF0C351gsJP9ohJqMOWDVbvMcVpxutpPtVrma6pVCa
7Pim37gAAK0IeQe+edCGtmZIBCIiJBIImJtEdJUeUKjGlc/p+nX37L/e+FaOaMvrHhHdUZ
OR1/fP1LB0y77opbF++zF28de2b1a3f+7HpB4Fz+/Mrbl1uGwNuGSxb1zRgaHZmxYDdNv
ouiP7zwdJApTsWlwomMl6ZHPnV2LEaQKDLBOUQ4byGu3t7+yr/nwlJNZ0Eg76ecvNcU
N+2ZR57exVAeaxvPjkgL6xACgYiINloriA1qhVqT1pqCZ8lZAQJ1Dczda3fdccTec+698Z5
NT6249RD7h21O+pbsvefen6k+95t179254moRjFvGRJu3DvX16E29h7z16l+O+8QhTV2b3t
E7KskZxx+4PR++/tIbIihFaeX9xtkTL+8z0dv0/dURr/v7yyoNxoXTF+aXn5jDZhnqhHLuJxq
zPE9sGXzGvGGkLtvHs5CJotgLK4HCBh+QOTAGpcQQJbGKFCE52jm2qwArrLPxbPD9
TfMP3P/wcz977KXnX/rAe62ehXvMmH3Hb55/+qlnibo5eAi5IVfP811jWxqiFvR179l8ccn+
H3pr7buR+HkzpkfjI6Vah+hw85UnnrTsE9P5A/PKT72/5wPXXL+2gdM3FNzT01GN0rxTb
SZT6nTlotwRwW3HFp62nrinTDsURvMZqj3tK/svzTMxSiulQCtR/6/zPxmBWuvJJxJqFq1
GXreudc01t15+04p/Xdq58bk1Ia/naeW3q6bv+v+eSS88/vjqv64lnhsRZQR9h3YYrvnfDyj+8X
mY55TOFWrlm19/X7/zkZ8666xcPT6uU1m3P3nhzI0XR8ud/ecOfH1kH29vliT3Sz1z/6EMt2
DispsYIVynMHQf2zTqsI5+jW5VWM6o2e7ev+BTMnj0UEL69auS43XfAlsFtJd02CACkFf
yvA6j/hSiKqFD59tZ4o9UWM3XJ4oXHXvtwb4e594ofPnfv1bud8sGDBwau/dlj37nwgo9+6
78HFHsviYluyvyrxxxyz17/cqjd+t68w/e79ZeP5Q17yqFLOhO55L67d5qJs/f+VE+6yEeWJ5q
hrf+y5Yr6vVctikpfP4RnvD/oxD2/coMMbArxWJmgZBulRRUJW81miJ+hm5c1nE/PftP4Oo
IT0opU6CqbJNVaa6U0ERljtNYadaJL1LLNVp61Xfu6S079lw8fpqWaPPDrR8678soWJdO
mlEvp+18/85ic65dceueNy/9LUBxr0yiUuk648t7pbv3baf6Fi264/OLzv/Ht7+8xrb9vFs6M482
GrvrNbScu7a0l8/b+wH86qLUUvJy86prq1TfcsdXv3TUjGMPWFB/PtdJL1Yg3bNDz+h+b
27Dx7ImnwEO170WTx3v4hyVABFFkY5iFSe6VI5KpZLWWimVgtJal6LOWnUx5TY2Ntj
N41s5zesLZ+0xvnlzeWp11W/fuur+17qiV6Z10K76yIZ1799416PXXfvFLTubPzjrSO6rrl63e
iTnmzmi0Do1qH+NA+YPHz2kO11mpH//lbwuXzcwVvR71926ZeHdYjc5fsHOPV1x9cMrfn
jmlzd+FH53GOz6U9xTKUYnhnZOzNKAu8P0kVDX/idj4z6fE+ec6ZK4QhOC8kISxXFXK
TVAlx1VrcWLv/WHl4QIL/zmNPD906odHfVpN97yqzZ6CNFNV13ylyce/2jf0Ff+0OqsTL3
y+OSYH73ejfjrn1w25s1nv/z9fRemakAP7uj/ykLf5Ua+7i/vnXryEc+++bv9j/7PG39wfj1vTx2
YVuqKJW/S0MTo/M3VMixamp6/WwwzxmlD9Z2/2gUldP05vIfJHPFdsO6/u5ae38lxtfrlbPe
9eiZU1+cfHTK6O2c/+WJmjNEqPXquP3pgazow8Ni2rX9dMxMUUZFTu1WM18eu+PFvv
nfekf0Gu+PKow8+/PJ7m9vRxI9Pmf3tT08zu1b94LOfPeuwJaPxjOsu/abDidriOC7L2tWr8uF
NI9780yP/deGln/vyyl2rBtc8b6mNzTC0dWdr+8TgjsFjjjooa5vtWyBsyOqOcaWdz4WnluYD
6NZpfNBm7qqBOth4CTrbNKAboO0vXN9PcrW/eJE9AQaNRhTilWKcue/+uNrGfROXX5
38vya3QtHzEB5y1jLTZt/8OCO2t77tSiq9sXnXPDZt7YM9uh0He510nUvVXrn3/HLB5c//X
Zp52uL9z/8sv862SRxrTzz1OMOrHZOzUV3UjbjmGpZ5NnnHok1sgrlt2+bdF4CK9/cXPviJ
waxbs1oZRFBJEMez2hIXHqLCI20B/vavuWynVLIVxbQPX6OruhdUDVuleKiLo5mLjrpv0
3RvWNJp6w07qGq3EInkwqFBNbZ23uQfcv6YgpaWZw4cVnN0Pj21d84Wv3ZUsPmd/Z2f

M8L1hbb/7bWQ/2x42pe3X8bf2bTuiNt3f+60F73HTX7++54MRZWlZBPujzhq6jEiprUkwq
xrh16uLmDR+e+fFPdB6+P+VEzlt5FcADewEdur8ezZoH7Zbp6Ue1QPUuotoBsOBCNeuT
MWTcnSVBbHdX/KNPbK0t6LMVu2Oq1+3qtafzD06f2GeKwTjQpOK21mqTvPX2rv6a+f
6Xr6y78NMfPXHNii9BZj8yPzqJhwRoauO91e3KurFNk0VECsWOV23YXDI0MXNKCSt1
pqEwEicmShwpUezaVJPxNfp9u+0Pi9tPog+5HwHJMaBHFGqX42q9d2/omsFqGbp2W/Wj
qRmJg59TuC4YVmtv/Li++tPbS9uHoB7DjNnTyi1oPdvc+DLPXXfQB9JIEqp2SnetVI6SJx9
adcudt597wtHtUteKa+7KwsTE1jzjuDOp/vpnL8XAq/560y6xFVP1oj3QxM6hiMPe0/sWleU
/VtzuCrIutAsc8Y1YxWnFeNRpcNpjuzkxV6+Lm33ZdYCvgVto1LDCAFLPoWWmzAMZ
EEYwbfAYQDsgCOPAs9I7j3Q9zc3e9kBH98SWrfnfNzWf744Pml4+stJ4b59bns5y1lQuRT
29abUS52PIHy7//Oz9lwThtSPFrmxi65ubX/vryhO/cOYwwx+XgdPNZhgvRyUDWli1JLcY
XXbSv1zx7Z9kJgdH4EhMdMF+R3PMUQQ+SQTHdgUzc1Fl6rujEzNr9bJc7Z4QYURa
MehaLtsO+rI0RjZFvgdAKOJ3UQ6USLS2QXQCvELm+H98VKOaYXKFzT1+MRTd/df8li
RW8mt6FrJl1NA0F/89j5/2/G62zgjc/76s4+/9YGHX9y0fqLdPPgDy+4P8OBwctFlN82Cy4qi
03XskvGuWIXh9vxpsOHQOc7awAEt63rY8M76VLjQWmEohKGJ7URPc+8O4/zYrS8Mlu
MeNdK0L0B5ic7f9eWsBG3H4nEqZ1vA9eUTa6q9iyQbbKm3IAR79y8bSPjIY7O4c2Nt74
Z5eXuoEmBsuHfVNYqTirtUpJ6V3JWrrr/p2lambL//IumfuU/r/3Jlz/18YE50SEnLTvxYyflL
GWUsi53ddY7RisvJcWViVlLn4rY3ZOKI2WDA/9e7qjimqAHIEOfur1yffnN8SSrG/Hsp
Rjy3boVHfMMVmpgGqiVtaePAeM+VDSaSNL3OnmwNHDxl+RWbfzfY0dU29aqHuB7H
aVxYwq44jVxhg0cU8uwAAK/47pLuPh3pKmGcZVklmYc26fTx1791Z4uSg5ccdM45CzrS
3sKYLLAIAUFeFMPNkbHGxOhI/fW33221Ws4F5a1GikutAjPsrHS0VQPS0Mz2bTe+f2qf
XqjV4EjY5UwzQNuMNF3PXjM3vzoWBsMmBMcUr1QPq6Uf3jZRagEzcSBQREGITKxD
kiSlpExECDr3oVmlAkECpZTOssz72CghikpJjQMmEI0H19Pf2wXVs05LIPiM20bVFCgA
RUoJYNXXCmd9RWbOnLlm7TvWO+9AMSdMHZvggJUrVxbTeg8qJr75MZvM1XHVh2
0GRzz6kGWQDms5tLb2lf6NjebDu0rzSsN5T60XWrXeNN3ccqHCjMELsotQGRSFIMEH1
1JRBBIUa7HeQkCQOI513gy+0Q6mApqpKCGQ40AMXzr7EKNDBBGQd1jYYlzHnYJG
GyMQJcZ2RrVgQm+l0tPV25zYGqESktT4x15757SKfO2DE2p+B1TqkBTXL/SOMWwZG3
mfTU9n7Xx7pexUE0NQnjevdCgMbigapWq4+MWdSkXWBxCxAVPROqZYcZKiUqDQI4
gW5Tkj8SEgkAgr3cqwlYNU9I0CcEyeyMOvI98QA2BGLgASBWl1mWCwQfRLCiljDHLc
tk0mmaKqVYRAB8I1l1Wkdzfreqvg+KYZxhtOBhDW0R9NGi3cxrbz00q2IwDdfUdsHI
VejN8Uzj6Qtm1EDCIgOXgBA/3P8TGIQCJoQRBAaQEAQDi6waIoAQNcbwQwXmW8h5
graxK0kLkUBIXfK+ulxTYS3aUUbNibKnYAEImo22977SXEEAHEcu+BF5AO6+aHHN5+
6sPuCRcCtcWwqTqO4u7NVa6Su+cp77747szsZNDQno57KyJo2QlyO/TtUKsbbiNo79G5y
hI4DW6UjbRBBhCdnJcAHb7MMPAaOADwR6fqEYU7yzHtAbxtFmCCiDmMqMXVXS
XOOI6jpKg7RQY7mRUiR3GSBNPKvS8ssJTjqFSO81y893N6Oq85aiSZtWh449qu9nhWD
rqzLEU72VlcWw5LyyEN0rYTIURcAo71aJYIJdmuZAqrzqrZOcykACkg9EQaUQBCAwC
gigC1jpCrRWLc140Iet2U7yl+jgJAiKMTKrigouMQdW6o1zC5B4hb2qfgxgAQIhSU0EFQT
ygK5eq06Z3Z822c+7MA4abwyaZ41RUwQ5W2yaer/sXABsEi8qmURbXGAs2zYfbWPH9
cyu7Xh8GDnFCmeFyCdSYQi1KKQ2gFDPbAKzQAABBIBctdRAfASVotZ1zIto6b51XG
pQy2qiATjEkJegsd5R1qhQppYUUKQVIgAKWRLN5Nmq9y7KW9z5SoWQSTATT0spxPr
RnW/a3997clTw45l3nosjkJc2xHjERGGYxykbcxEOVKnnCHbG2ECKnS3uRh6reBs5iREo0
sNZaQkCltAFjSIOAqBjBY4QAIEYAUQhBWxYAUAUToXMBPanY1zwmilOjJrsAWmsh
hUe2nEAQZ+7LMvyPJ/kdUmSiIi19t5tXXfsNNqOW1HzVEylLNIo6CNSbc7iiCAKkXJ2r
JqPFaYS9/RVZvj6kFd9PeqWd1TefHcnBxNFFBMRoEKJE51EWikVkwHRAGSD1xqI2AY
XELT1QAAeSEMwIh5dQogsxsSgGpuEtEEADhAIRVNkNAKDQ2+DdxxIYblkIqeVUo28
7fK87JOcSwbdUBQ6ESLlAF0aWzZqnEIMEfhaHpKJocbiruzOpbGt22TqFKCup/3OCMSY
QxTpSDERGVZxrGONsTEIWIEMjBgirUIAXy5CBqKdCyQghChaKwYHQF6JriWlqkmM
NqREKEwOb4oYIiQIWgUCFxsKAKaBQIIqISV1ZIABVeQ9QbC+4LYy1ud5HqYTKw0FR
pYwFODHzN4LMm+bsKBW1HeFthMw5y6Nfr7BKIWx1oqMZijHIEQm0tpgBKJZUDNaa
5kZKqVIwf8BOqghp0m4AkUAAAAASUVORK5CYII=", "text/plain": [""] },
"execution_count": 28, "metadata": {}, "output_type": "execute_result" }], "source": ["img=

```

image.load_img('/content/animals/animals/butterfly/16a79172c2.jpg',target_size=(64,64))\n",
"img" ] }, { "cell_type": "code", "execution_count": null, "metadata": { "colab": { "base_uri":
"https://localhost:8080/" }, "id": "68DbrPdbjuB", "outputId": "a29e5ff7-d650-4821-8436-
2afe9b60b38e" }, "outputs": [ { "name": "stdout", "output_type": "stream", "text": [
['antelope', 'badger', 'bat', 'bear', 'bee', 'beetle', 'bison', 'boar', 'butterfly', 'cat', 'caterpillar',
'chimpanzee', 'cockroach', 'cow', 'coyote', 'crab', 'crow', 'deer', 'dog', 'dolphin', 'donkey',
'dragonfly', 'duck', 'eagle', 'elephant', 'flamingo', 'fly', 'fox', 'goat', 'goldfish', 'goose', 'gorilla',
'grasshopper', 'hamster', 'hare', 'hedgehog', 'hippopotamus', 'hornbill', 'horse', 'hummingbird',
'hyena', 'jellyfish', 'kangaroo', 'koala', 'ladybugs', 'leopard', 'lion', 'lizard', 'lobster', 'mosquito',
'moth', 'mouse', 'octopus', 'okapi', 'orangutan', 'otter', 'owl', 'ox', 'oyster', 'panda', 'parrot',
'pelecaniformes', 'penguin', 'pig', 'pigeon', 'porcupine', 'possum', 'raccoon', 'rat', 'reindeer',
'rhinoceros', 'sandpiper', 'seahorse', 'seal', 'shark', 'sheep', 'snake', 'sparrow', 'squid', 'squirrel',
'starfish', 'swan', 'tiger', 'turkey', 'turtle', 'whale', 'wolf', 'wombat', 'woodpecker', 'zebra']\n" ] }
], "source": [ "# opening the file in read mode\n", "my_file =
open('/content/90_classes_name_list.txt', 'r')\n", "\n", "# reading the file\n", "data =
my_file.read()\n", "\n", "# replacing end splitting the text\n", "# when newline ('\n') is
seen.\n", "i = data.split('\n')\n", "print(i)\n", "my_file.close()\n" ] }, { "cell_type": "code",
"execution_count": null, "metadata": { "colab": { "base_uri": "https://localhost:8080/",
"height": 53 }, "id": "Jc-sSICvUPpK", "outputId": "abba7e43-ae9e-4f90-aba1-
0165c4a75864" }, "outputs": [ { "name": "stdout", "output_type": "stream", "text": [ "1/1
[=====] - 0s 20ms/step\n" ] }, { "data": {
"application/vnd.google.colaboratory.intrinsic+json": { "type": "string" }, "text/plain": [
"butterfly" ] }, "execution_count": 75, "metadata": {}, "output_type": "execute_result" } ],
"source": [ "x= np.expand_dims(img,axis=0)\n", "x= image.img_to_array(img) # converting
the image into array\n", "\n", "x= np.expand_dims(img,axis=0)\n", "pred =
np.argmax(model.predict(x)) # predicting the higher probability index\n", "op1=i # creating a
list\n", "op1[pred] # List indexing with output" ] }, "metadata": { "accelerator": "GPU",
"colab": { "gpuType": "T4", "provenance": [] }, "gpuClass": "standard", "kernelSpec": {
"display_name": "Python 3", "name": "python3" }, "language_info": { "name": "python" } },
"nbformat": 4, "nbformat_minor": 0 }

```