{ "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\":\"c208fb742e7909d1ce586ebdebfc06a7\"}'}" ] }, "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,validation\_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,validation\_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": "", "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": "68DbrPdbjjuB", "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-sSlCvUPpK", "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 }