Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct

objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a

neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other

objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension.

So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more

thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh

use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image

one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification

here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and

thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural

language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw

data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description

to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these

classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2

okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see

here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So

if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you

this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain

this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine

learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model

trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it

has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in

collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning

model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset

which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our

previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip

file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image

the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand

meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and

other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine

learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a

model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered

explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In

today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they

have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say

resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this

flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are

image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made

model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If

you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll

now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a

model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead

in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the

steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained

model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have

imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a

classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I

have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's

why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and

let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by

Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's

see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the

topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and

apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh

go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes

see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index

you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to

say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our

image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time.

It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different

domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even

weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and

improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's

get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training

none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among

those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels

okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images

okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then

I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive.

When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer

vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large

label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer

series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the

weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then

you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it

224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just

say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of

the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0.

But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN

previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and

machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider

using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is

written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they

are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one

big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225

4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9

is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder

and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier

to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters

are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power

because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen

layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done

some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a

bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here

and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one

is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous

layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:one of the most powerful ideas in deep learning is that sometimes we can take the knowledge the new network has learned from one toss and apply that knowledge to a separate task so for example maybe kind of a new network learn to recognize objects like cats and then use that knowledge or use part of that knowledge to help you do a better job reading x-ray scans this is called transfer learning let's take a look let's say you've trained in your network on image recognition so you first take a neural network and train it on XY pairs where X is an image and Y is some object in the image as a cat or a dog or bird or something else if you want to take this new network and a

gap or we say transfer what is learn to a different tasks such as radiology diagnosis or meaning really reading x-ray scans what you can do is take this loss output layer of the neural network and just delete that and delete also the waste feeding into that loss output layer and create a new set of randomly initialized ways just for the last layer and at that now output radiology diagnosis so to be concrete during the first phase of training when you're trading on an image recognition task you train all of the usual parameters within your network all the ways all the layers and you have something that now learns to make image recognition predictions having trained that neural network what you now do to implement transfer learning is swap in a new data set X Y where now these are radiology images and why are the diagnosis you want to predict and what you do is initialize the last layers ways is called a WL + BL random being and now we train the neural network on this new data set on the new radiology dataset we have a couple options of regions in your network with radiology data you might if you have a small radiology data set you might want to just retrain the weights of the last layer just W LPL and keep the rest of parameters fixed if you have an up data you could also retrain all the layers of the rest of the neural network and the rule of thumb is maybe a bit of a small data set then just retrain the one loss layer at the output layer or maybe even lost one or two layers there's a lot of data then maybe you can retrain all the parameters in the networks and if you retrain all the parameters in your network then this initial phase of training on image recognition is sometimes called pre training because you're using image recognition data to pre initialize or really pre train the weights of the neural network and then if you are updating all the ways afterwards and trading on the radiology data sometimes that's called fine tuning so if you share the words pre training and fine tuning in a deep learning context this is what they mean when they refer to pre training and fine tuning ways in a transfer learning cost and what you've done in this example is you've taken knowledge learn from image recognition and applied it or transferred it to radiology diagnosis and the reason this can be helpful is that a lot of the low-level features such as detecting edges that encourage detecting positive objects learning from that from a very enlarged image recognition data set might help your learning algorithm do better in radiology diagnosis it's just

learned a lot about the structure and the nature of how images look like and some of that knowledge will be useful so having learn to recognize images it might have learned enough about you know just what parts of different images look like that that knowledge about lines dots curves and so on may be small parts of objects that knowledge could help your radiology diagnosis Network learn a bit faster or learn what less data here's another example let's say that you've trained a speech recognition so now X is inputs of audio or your snippers and Y is some in transcript so you're trained in speech recognition system to output your transcripts and let's say that you now want to build a waik word or a trigger word detection system so recall that they wake whether the trigger words are the words we say in order to wake up speech control devices in the houses such as saying Alexis and we're going Amazon echo or okay Google to waken Google device or a series with an Apple device or saying they hope I do to wake up up my to device so in order to do this you might take out the last layer of the neural network again and create a new output note but sometimes another thing you could do is actually create not just a single new output but actually create several new layers to your neural network to try to predict the labels Y for your wake word detection problem then again depending on how much data you have you might just retrain the new layers of the network or maybe you could be trained you're even more layers of this neural network so when does transfer learning makes sense transfer learning makes sense when you have a lot of data for the problem you're transferring from and usually relatively less data for the problem you're transferring to so for example let's say you have a million examples for your image recognition tasks so that's a lot of data to learn a lot of low-level features or to learn a lot of useful features in the earlier layers in your network but for the radiology tasks maybe you have only 100 examples so you're very low data for the radiology diagnosis problem you have only 100 x-ray scans so lot of knowledge you learn from image recognition can be transferred and can really help you get going with radiology recognition even if you don't have an all the data for radiology or speech recognition maybe you've trained the speech recognition system on $10,000 of data so you have learned a lot about what human voices sounds like from that $10,000 of data which really is a lot but for your trigger word detection maybe

you have only one hour of data so that's not raw data to figure out parameters so in this case a lot of what you learn about what human voices sound like what are components of human speech and so on that can be really helpful but building a good wake word detector even though you have a relatively small data center he's a much smaller data set for the weak word detection task so both of these cases are transferring from a problem with a lot of data to a problem with relatively little data one case where transfer learning would not make sense is if the opposite was true so if you had a hundred images for image recognition and you had a hundred images for radiology diagnosis or even you're a thousand images really for radiology diagnosis one would think about it is that to do well on radiology diagnosis assuming what you really want to do well on is radiology diagnosis having radiology images is much more valuable than having cat-and-dog and so on images so each example here is much more valuable than each example there at least for the purpose of building a good radiology system so if you already have more data for radiology is not that likely that having 100 images or your random objects of cats and dogs and calls and so on would be that helpful because the value of one example of images from your English recognition terms of cats and dogs is just less valuable than one example of an x-ray image for the task of building a good radiology system so this would be one example where transfer learning well it might not hurt but I wouldn't expect it to give you any meaningful gain either and similarly if you built a speech recognition system on 10oz of data and you actually have 10 hours or maybe even more say 50 hours of data for week word detection you know it won't merely not hurt maybe it won't hurt to include that 10 hours of data to do transfer but you just couldn't expect to get a meaningful game so to summarize when does transfer learning make sense if you're trying to learn from some task a and transfer some of the knowledge to sometimes be then transfer learning make sense when toss a and B have the same input X in the first example a and B both images as input in the second example both had audio codes as input it tends to make sense when you have one more data for toss a then toss B all this is under the assumption that what you really want to do well on this toss video and because data for tossed B is more valuable for toss B usually you just need a lot more data for toss a because do

each example from toss a is just less valuable photos B in each example for toss B and then finally transfer learning will tend to make more sense if you suspect that low-level features from toss a could be helpful for learning times B and in both of the earlier examples maybe learning image recognition teaches you a number about images to hover radiology diagnosis and maybe learning speech recognition teaches you about human speech to help you with trigger words on record detection so to summarize transfer learning tends to be most useful if you're trying to do well on sometimes be usually a problem where you're relatively little data so for example in radiology you know difficult to get that many x-ray scans to build a good radiology diagnosis system so in that case you might find it related by different tasks such as image recognition where you can get maybe million images and learn a lot of low-level features from that so that you can then try to do well on toss be on your radiology task despite not having damage data for it when transfer learning makes sense it does help the performance of your learning algorithms significantly but I've also seen sometimes seen transfer learning applied in settings where toss a actually has less data than toss B and in those cases you kind of don't expect to see much of a game so that's it's a transfer learning where you learn from one toss and try to transfer to a different task there's another version of learning from multiple toss which is called multitasking which is when you try to learn from multiple tasks at the same time rather than learning from one and then sequentially or after that trying to transfer to a different task so in the next video let's discuss multitasking

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube

channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural

network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second

one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and

achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm

performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code

from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get

the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know.

So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by

255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then

I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset

and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial.

So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know

they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am

going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are

getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in

enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do

not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to

use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want

to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see

some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving

forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its

original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation

enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new

problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the

layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow

installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one

image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me

the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so

that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last

layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of

an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer

and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at

different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some

deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am

going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are

adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is

converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you

know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one? So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last

layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!

Text:foreign do you know training a classifier to distinguish beverages can help predict the cuisine of an image to know more about cancer learning and how it works stay tuned till the end of this video in this video we will cover topics like what transfer learning is how transfer Learning Works moving forward we will dive into why you should use transfer learning after that we will cover the steps to use transfer learning and at the end we will see popular model trained using transfer learning let me tell you guys that we have regular updates on multiple Technologies if you are a tech geek in a continuous hunt for the latest technological Trends then consider getting subscribed to our YouTube channel and press that Bell icon to never miss any update from sip leader by the end of this video I can ensure that all your questions and doubts related to transfer learning will have been cleared also accelerate your career in Ai and ml with our comprehensive postgraduate program in Ai and machine learning boost your career with this Ai and ml course delivered in collaboration with party University and IBM learn in demand skills such as machine learning deep learning NLP computer vision reinforcement learning generative AI prompt engineering chargedy and many more you will receive a prestigious certificate and ask me anything session by IBM with 5 Capstone in different domains using real data set you will gain practical experience master classes by Buddy University and IBM experts ensure top-notch education simply learn job assist helps you get noticed by Leading companies this program covers statistics python supervised and unsupervised learning NLP neural network computer vision Gans Keras tensorflow and many more skills so why wait enroll now and unlock exciting Ai and ml opportunities the course Link in is in the description box below so without any further Ado let's get started so what is transfer learning transfer learning is machine learning refers to G using a pretend model to improve prediction on a new task it involves using knowledge gained from a previous assignment to tackle a related problem for instance a model trained to recognize backpacks can also be used to identify other objects like sunglasses due to the substantial CPU power required this approach is widely utilized income Division and natural language processing tasks including sentiment analysis so moving forward let's see how transfer

Learning Works so how does transfer learning work in computer vision neural network have distinct objectives for each layer detecting edges in the first layer and identifying forms in the middle layer and capturing tasks specific features in the later layer transfer learning utilize the early and Center layers for a pre-pretent model and only retrains the later layer it leverages the label data from its original task for instance if you have a model trained to identify backpacks in images and now want to use it to detect sunglasses we will retrain the later layers to understand the distinguished features of sunglasses from the other objects so moving forward let's see why should you use transfer learning transfer learning offers several advantages including reduced training time improved neural network performance in most cases and the ability to work with limited data training a neural model from scratch typically requires a substantial amount of data which may not always be readily available transfer learning becomes valuable in such scenario here is why you should consider using transfer learning first one is efficient use of data with pre-trained models you can perform well even with limited training data that is specially beneficial in tasks like NLP where obtaining large label data set can be challenging and time Computing the second one is faster training building a deep neural network from a scratch of a complex task can be time consuming taking days or even weeks by leveraging transfer learning the training time is significantly reduced as you start with a model that has already learned General features from a related problem now moving forward let's see steps to use transfer learning the first one is training a model to reuse it in machine learning training a model involves providing it with the data to learn patterns and make prediction once a model is trained on a specific task it can be reused and repurpose for related tasks saving time and computational resources the second one is using a pre-trained Model A pre-trained model is a model that has already been trained on a larger set for a specific task instead of training a model from scratch using a pre-trained model as a starting point allow us to benefit from the knowledge it has gained during its previous training the third one is extraction of features feature extraction is a process in which meaningful patterns and characteristics are identified and separated from a raw data in the context of machine learning it involves identifying relevant information from input data to

feed into a model for a better prediction the fourth one is extraction of features in neural networks a neural networks feature extraction involves identifying important patterns or features in the data at different network layers the early layers typically capture simple features like edges while deeper layers capture more complex feature relevant to the task at hand this hierarchical representation enables neural network to learn and generalize from the data effectively so moving forward let's see some popular models trained using transfer learning so numerous machine learning models have been trained using transfer learning some popular ones include for the first one is vgg16 and vgg 90. these models were trained on the image net data set for image classifications test the second one is inception V3 these models were pre-trained on imagenet and are known for their effectiveness in which rapid in object detect repeat in object detection and object recognition the third one is bird bi-directional encoder representation from Transformer this language model is written on the extensive text collection and find extensive application in NLP tasks like sentimental analysis and name entity recognition the fourth one is GPT generative pre-trained Transformer series these models are printed language models for various NLP tasks these are just a few example of pre-trained models that have been used in transfer learning to accelerate training and improve performance across different tasks and with that we have come to end of this video on what is transfer learning I hope you found it useful and entertaining please ask any question about the topics covered in this video in the comments box below our experts will assist you in addressing your problem thank you for watching stay safe and keep learning with simply learning staying ahead in your career requires continuous learning and upskilling whether you're a student aiming to learn today's top skills or a working professional looking to advance your career we've got you covered explore our impressive catalog of certification programs in Cutting Edge domains including data science cloud computing cyber security AI machine learning or digital marketing designed in collaboration with leading universities and top corporations and delivered by industry experts choose any of our programs and set yourself on the path to Career Success click the link in the description to know more hi there if you like this video subscribe to the simply learned YouTube channel and

click here to watch similar videos turn it up and get certified click here foreign

Text:Transfer learning has become quite popular in the field of image classification and Natural Language Processing. Here we take a pre-trained model and then we try to retrain it for the new problem. So if you remember from our data augmentation tutorial, we had flowers dataset where we are trying to classify five type of flowers. So in this video we will use a Mobilenet pre-trained model from Google's Tensorflow hub and we will use that pre-trained model to classify our flowers dataset and you will see that previously it used to take many epochs to train the complete model and achieve high accuracy. In this case using a pre-trained model it takes only like two or five iteration or epochs to get a superb accuracy. So using transfer learning saves lot of computation power because many times these pre-trained models that you can get from places like Tensorflow hub they are trained on millions of images. If you try to train that model on your computer it might take days or even months. But all you're doing is you're taking that pre-trained model, you're getting all the weights and everything and then you kind of change only the last layer or last few layers for your new problem and then uh you can get a superb high accuracy uh with this kind of approach. So let's get started we'll go over some theory and then we'll uh do coding. This is Wikipedia's definition of Transfer Learning which is you focus on storing knowledge gained while solving one problem and apply it to a different but related problem. For example if you have a model that can recognize cars it can be used to recognize trucks as well because the basic features, for example the tires, the steering wheel and some of the components between cars and trucks will be still similar. So you can use this knowledge of this visual world to transfer that knowledge into solving a different problem. In today's coding problem what we are going to do is we will take a Google's trained Mobilenet V2 model which is trained on 1.4 million images and total thousand classes. So this is a deep learning model that is trained at Google it would have taken a long time and a lot of computational resources you can see 1.5 4 million images is pretty huge dataset and the output is 1000 classes and these classes are little diverse you know. You have a goldfish, shark, some animals then some

Hammerhead military uniform so you have it's not just the animals it's animals and some other objects total thousand classes and when this model is trained it will have input layer, then some deep layers and hidden layers in between then in the end you have a softmax layer which is just you know classifying it into thousand categories. In deep learning what happens is we freeze all the layers except the last one. So you know all these layers that you see, we will freeze it and then we'll use this model to classify flowers which could be one of the five flower types which I have shown here and we are going to use same dataset that we use in our data eggman augmentation tutorial. So when you freeze this layer what happens is the model weights don't change. So now when I'm performing my training, so by the way you take the model and then you still have to perform the training that's very important. But when you're performing a training the the weights in these frozen layers are not changing. So it almost you know looks like a con equation. So you are having this one big non-linear equation so you are passing your flower and this is a training phase, and then during using this weight you will get a feature vector. You are almost doing a feature engineering and then you use soft mix to classify into five classes instead of thousand. So I hope you get an idea that you're almost generating the features feature vector using this frozen layers. So during the training none of the weights nothing changes okay, and omitting the last layer is a very common approach in some approaches they also freeze only let's say three layers or two layers and remaining layers uh go through the usual neural network training okay? So we're going to now uh do a python coding uh to use Mobilenet V2 model and then use it to classify the flowers. We will download a pre-trained model from a place called Tensorflow hub. So Google has come up with this Tensorflow hub where you can get an access of different pre-trained models. So right now for tax domain problems they have all these models, for example for embedding they have 176 models, for image classification they have 188 models. So these are like pre-trained models which are trained on a huge dataset which you can import and directly use it. For video also see they have some video and audio so they have some problem. So if I look at image classification here there is this model called Mobilenet V2 okay? So this is the model we are going to use so this model as I said is trained on 1.4 million uh

images and 1000 different classes, and the image is like 2224 by 224. you know it is that dimension. So now here in my jupyter notebook I have imported all essential libraries, and the first thing I am going to do is create is basically import that Mobilenet V2 classification model. So this is how we import it. So I have imported Tensorflow hub now this doesn't come with your regular tensorflow installation you have to install it separately. So make sure you run pip install Tensorflow hub otherwise it will give you model not found error. Here I am creating a classifier directly using this particular Mobilenet. So if you look at see so you have to give this this particular string or you know they have a code snippet. So you just copy that and by the way I have uh used some of the code from Tensorflow uh official tutorial. So thanks Tensorflow credit goes to you. But I have made it little simpler, you know so I have I have omitted the things which are not needed. So it is I have done some customization, so now here the image shape you know the image shape as you saw was 225 4 by 224, so you need to give two to four, two to four and I'm adding the third dimension for the channel. So what happens is when you do this in numpy okay let me just import it it will just make it 224 by 224 by 3 okay? So whenever it comes up you see that so that is the input shape I am giving and once you do that see you have the the model ready only. So now if you want to classify among those thousand classes okay, so let me open the file that I have so here in my current project directory I have downloaded the those thousand classes and if I open that file these are the classes see total thousand classes and uh goldfish is one of the class. So I'm like okay let me try to classify goldfish. So I downloaded goldfish picture and I'm going to use this model to classify that. So I have imported a pillow model and image from that, and you know you can just say image. open the file name is gold goldfish this is how the image looks but we have to resize it to 224. So I will just say resize to image shape, and I will just store it here okay? So it's a smaller image now and let me try to classify this image. So now before you classify it you have to uh scale it you've seen in all of our previous tutorials that before giving it for classification or training, we always scale or normalize the image and how do you do that. Well the the color scale is 0 to 255 so you divide it by 255. So see here I'm dividing it by 255 and when you do that uh the value of goldfish is like like if you look at just

this array, see now these values are in between zero and one range okay? I'm gonna do one more thing which is see when you do something like this, what you're doing is you are changing you are adding uh one more dimension which is one, and the reason I am doing it is because when you do prediction you know prediction accepts multiple image as an input. You cannot have like only one image as an input so that is the only reason I am doing it. So now I can do classifier predict like this so now uh you have a total thousand classes okay? So this is making a prediction for each classes, each class like zero classes this probability one class has this probability and so on. So here I am going to store this in less a result and let's look at result.shape it's thousand okay? Now I need to get the max. So when you do np .arg max from result it will give you the value the index which has a maximum value and if you notice previously say it's very upfront 0,1,2 see this has a this has a bigger value at least in this view 9. So that's what it is giving you. Now how do I know which class 9 is? Well uh if you look at the just a second if you look at our image labels two classes goldfish okay, so it's very clear but just to make it proper here what I will do is I will uh open this file so I will just say with open okay with open what well this particular file as f and f.read will read the files and when you do split lines it will split the lines, and you want to store this into an array called image labels okay? So image label is nothing but a label array and if you look at these labels you will find that now you are having those thousand classification labels, and if you supply your predicted label index you get a goldfish here. So this looks good so far. We used pre-trained model and we just did classification straight away this is like you know almost loading a pickled model and doing a classification. Now we want to do a classification for our flowers dataset and you can download flowers dataset by running this command. Now we have done data augmentation tutorial on this flower dataset before in the previous video and majority of the code is same. That's why I'm not going into too much details. But if you check this code here all you're doing is downloading this zip file of all the flowers from Google website this is how you you you download it and if you look at data directory the directory is dot means local directory that has data set folder that has flower photos. So if you look at our folder see dataset folder has flower photos and that has all five flowers. So daisy

will have daisy flowers see daisy will have daisy, roses will have resist flower and so on. So let's uh use a pathlib direct path lib python module to convert this path this is a string path all I'm doing is converting it into windows path. Now why am I doing it? Well so that I can do functions like this so when I have Windows path and if I do star.jpg it will go recursively into all the directories and get me the file path of individual images, and those paths will be needed and if you look at image count we have these many images and now I am going to get all the roses images so from data directory I am saying go to roses folder, roses folder and star means get me all the files, and that file path you are getting in this roses' is our directory our roses are list okay? Let's try opening some files you know. So I'm using this image is a pillow library so you can use this code to open first row's image, second row's image and so on see similar thing you can do with Tulips. So if I let's say supply Tulips here and what is this Tulips? Tulips is the name of the folder you see Tulips here is a name of the folder and if you do that you get all this Tulips and if you open some Tulips images Tu lip s so you get all this beautiful looking images. Now I'm going to conver make a python dictionary so that the key of the dictionary is the flower name, and the value is the list of images. So in this dictionary now if I do roses this will give me a pile path of all rose images. Similarly Tulips gives me all Tulips images okay? We have we have seen all of this in previous videos so you should be aware and I'm creating a label directory as well because machine learning module doesn't understand text. So you need to say okay roses is 0, daisy is 1 and so on. Now if you um look at let's say any particular file path it looks something like this, and this you can now read into opencv. So cv2 is opencv model which I have imported at the top, and I am saying iamread which means image read and this thing is same as this. So I'm let's say reading one image and if you look at image path you know image paths are image shape sorry image shapes are different. So I need to resize it because I want to in before training your model you need to make sure all images are of same size. So here I will make image the same size see this is how we do it. So now I will run a for loop on my dictionary and create x and y this is something again we did in the previous video that's why I'm not going into detail. But if you this code is very simple you're going through your this particular dictionary, for each rose you're

going through each images. So going through each image is a second for loop, then you read image one by one then you resize it and you append the resize image to x and you append the label you know to y. So if you look at x of zero it's a three dimensional array of between 0 and 255. But we saw in previous videos that before doing image classification training we have to divide it by 255 so that it can scale. See if you do that it will bring it bring the value to 0 to 1. And if you want to do it on the entire dataset this is a numpy is so convenient, you can um convert first into numpy array then we'll divide it into 255. So let's do train test split first. This is a standard code we have seen in enough video so it doesn't need any explanation, and then we can divide it divide these images by 255. So when I look at this thing you know it's it's in this range 0 to 255. Now I want to use that pre-trained model and classify some of these images. So let's say first one is daisy, the second one is a beautiful rose, the third image is let's say again it's another rose. So let's try to use our classifier to predict this model. So this classifier is what? Well we saw previously, it is our pre-trained model that we imported from Tensorflow hub you know ready-made model, and I can now predict x of 0. But you know this takes numpy array. So you have to give numpy array I will I will give x of 0, x of 1, and x of 2 okay, and it return this um array of predicted arrays. So I will store it in predicted and then I can do an argmax arg mix will give you the maximum argument and what it is saying is the first flower this flower is 7 9 95 this flower is 880 the third flower is 795. So what is 795? Well we had our image levels remember in that if you supply 795 it's saying this is a flower curtain. Maybe on Mobilenet when when Google trained it maybe if some shower curtain had this flower pattern that's why it is saying. Even 795 that this image is also saying it's a flower curtain and 880 what is 880? So what is this? Oh this it is predicting as umbrella. So you see you cannot here use your ready-made model because ready-made model only has daisy as a flower. It even doesn't have all these or four different flowers. So it's gonna make some random guess out of those thousand classes, thousand meaning all these classes. And by the way this file and this notebook everything is in available in video description below. So make sure you download this from my Github. Now I'm going to retrain this model and here I have a feature extractor model. So how is it different than the previous one?

So previously if you remember look at it this whole path is same the only thing I have classification here, here I have feature vector. So this gives the same model as the previous one except the last layer. So if you look at our presentation you know this is the whole model but from that model you want to take only the layers which doesn't include the last layer, and all these layers excluding last layer is given by this feature vector. So you can now create a model like this: so again I'm using Tensorflow hub creating Keraslayer and passing this URL here input shape is standard shape, this is an important parameter. You are saying trainable false which means freeze. See freeze means do not train. So when now you perform a training all those layers will have their fixed weights, and then I can create my model like this. So I am putting that that ready-made model and then creating the last layer which is that the classification of five flowers. See, so only last layer is mine the previous layers are are already trained, and then I will run only five epochs by the way. So these parameters are standard Adam, Sparse category, cross entropy, etc., and I am now running only five epochs. Now if you remember from our data augmentation tutorial, to train the same model with CNN previously it took us you know 30 epochs. In 30 epochs we got 85% accuracy. Now check this- in second epoch you got 85% accuracy so you can see that deep learning training is very expensive. When you run so many trainings uh so many epochs your GPU, CPU your electricity power is burnt. You might get a big electricity bill, but with transfer learning you can save all that computation time. It is not just the bill sometimes when you're building a training a big model, it might take you days, weeks or months. But with pre-trained model you can retrain it for your problem so much easily and let's look at the the performance of our test dataset- that is also very good 85 percent. So this is the reason why transfer learning is so popular in computer vision and nature language processing. If you are solving any computer vision or NLP problem try to see if you can use transfer learning. If you cannot then only try to build the model from scratch. I hope you like this video the notebook and other links are available in the video description below so make sure you check it, and make sure you watch all these videos in this deep learning series. I have a separate tutorial series on machine learning python, I'm doing python projects as well nowadays so if you want to learn python or small

projects. I have a complete playlist on variety of projects as well. So make sure you check it out, and thank you for watching! Goodbye!