		ions  will use	, \	Dense									
Iı	ba "" nis function is a da nputs     train_data_path     val_data_path     targetsize     classmode     batchsize utputs	_	path for data pat e for the , 'catego train data	the data h for the generato rical' a	set datase	t			1)				
V:	rain_datagen=ImageDa al_datagen = ImageDa rain_generator=train al_generator=val_data	taGenerator(p	reprocess	ectory(tr target_ color_m batch_s class_m shuffle	ain_data_size=tanode='rgaize=batanode=clase=True)	a_path, ; rgetsize b', chsize, ssmode, , # this	input) # this is wh , is where yo						<sup>F</sup> older
	e <b>turn</b> train_generato rain_evaluate_the_mo	,	erator, \ ator, \ , \	color_m batch_s class_m	size=ta node='rg size=bat node=cla	chsize,							
Ot .	rain the model, do proputs train_generator: val_generator: optimizer: epochs: dropout_value: TheModel: utputs accuracy:	generated tr generated va the optimize the epochs o the dropout The pretrain	nd do eva ain data lidation r method f the mod value ed model	data used to c el use to loaded, odel	compile o	the mode	·	he model					
X X X X X pi mo	= TheModel.output = tf.keras.layers.F. = Dense(1024, actival) = Dense(512, actival) = tf.keras.layers.D. redictions = Dense(4) odel = Model(inputs)  freeze base layers or layer in TheModel layer.trainable =	ration="relu") ration="relu") ration="relu")( ropout(dropou r, activation= TheModel.in  for traininglayers:	(x) (x) x) t_value)(: "sigmoid"	x) )(x)		s)							
si # r	metrics=['a tep_size_train=train fit the model = model.fit_generate va	orical_crosse ccuracy']) generator.n/ or(generator= alidation_dat teps_per_epoc pochs=epochs)	/train_gentrain_gent a=val_gent h=step_si	erator, erator,		ze							
train_val_da	d and gene  datagen=ImageDataGen  atagen = ImageDataGen  atagen = ImageDataGen  al and constant parameters batch_size : the	rate dat enerator(prepr enerator(prepr	ocessing_ ocessing_ e model	function=	preproc	ess_inpu	t)	in our	dependenc	ries			
# constarged class of the class	ant parameters target_size : Co	enstant image lass mode we w (4,224) egorical'	target si	ze for th	ie whole	noteboo	k which is (						
train_validatest_c	a paths _dataset_path = 'Data ation_dataset_path = dataset_path = 'Datas _generator_initial = enerator_initial = va	: 'Dataset/Val set/Testing/' : train_datage	idation/' n.flow_fr	target_ color_m batch_s class_m shuffle irectory(	size=ta node='rg size=bate node=cla e=True) validat	rgetsize b', chsize_i ssmode_c ion_data:	_constant, nitial, onstant, set_path, #				·		
Found Found # disp train_	2870 images belongin 2870 images belongin 394 images belonging olay classes found in _generator_initial.c. ma_tumor': 0, 'menin	ng to 4 classes to 4 classes n the data class_indices		color_m batch_s class_m shuffle	node='rg size=bat node=cla e=True)	b', chsize_i ssmode_c	onstant,						
image_ # impo efnB7	censorflow.keras.app.  _size = [224,224] # cort the base model = tf.keras.applicat.  ring the base model _basemodel = efnB7	choose image	ort Effic size tnet.Effi	ientNetB7 cientNetB	, 37(input	_shape= :		3],weigh	nts='image	enet',incl	Lude_top	=False)	
# init dropout  # add  x = et  x = De  x = De  x = De  predic	rers on top of the bas tial parameter value ut_value_initial = 0 layers fnB7.output f.keras.layers.Flatte ense(1024, activation ense(512, activation f.keras.layers.Dropol ctions = Dense(4, activations)	en()(x) on="relu")(x) on="relu")(x) on="relu")(x) out(dropout_va	<pre>moid")(x)</pre>										
'\n# a x = De = Mode # free for la	= Model(inputs = effect of definition)  dd layers\nx = efnB7 nse(512, activation= l(inputs = efnB7.inp eze base layers for ayer in efnB7.layers ayer.trainable = Fals  npile the model	<pre>7.output\nx = ="relu")(x)\nx out, outputs = training :: .se</pre>	tf.keras. = tf.ker	layers.Fl as.layers	Latten()								
# comp model "\n# cy'])"	tial optimizer izer_initial = 'Adam  pile the model compile(optimizer=options='category metrics=['adam  compile the model\nmo  he model	ptimizer_init orical_crosse ccuracy'])	ntropy',	optimizer	initia	1,\n	lo	ss='cate	egorical_c	crossentro	opy',∖n		metrics
epochs step_s  tic = # fit r = mo	S	denerator=trai validation_dat steps_per_epoc spochs=epochs_	n_generat a=val_gen h=step_si initial)	or_initia erator_in	al, nitial,		ial.batch_si	ze # con	nputing th	ne steps s	size per	epoch	
C:\Use will b warn Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Minute	rs\Student\anaconda_e removed in a futur ings.warn('`Model.fi 1/4 8 [====================================	_3\lib\site-pa re version. Pl lt_generator`	ckages\te ease use is deprec [] - 1367s [] - 1448s [] - 1494s	`Model.fi ated and 4s/step 4s/step 4s/step	- loss: - loss: - loss:	2.3466 0.4798 0.3648	rts generato - accuracy: - accuracy: - accuracy:	0.6600 · 0.8298 · 0.8676 ·	- val_loss - val_loss - val_loss	6: 0.4941 6: 0.2534 6: 0.3082	- val_a - val_a - val_a	ccuracy: 6 ccuracy: 6 ccuracy: 6	0.8362 0.9045 0.8895
# evalua scores print 359/35 evalua	luate the meduation on validation is = model.evaluate(valuate(valuate(valuate(valuate))))  9 [===================================	on data val_generator_ evaluate ",mo	del.metri	2s/step -	- loss:		• •	.9544					
param_param_accuratic = for pa	label = 'optimizer'  _label = 'optimizer'  _list = ['Adam', 'SGacy_table = {param_latime.time()} aram in tqdm.tqdm_no aram in tqdm.tqdm_no aram, and evaluate accuracy, _ = train_er  collect results accuracy_table[param_saccuracy_table['accuracy_table]	abel: [], 'ac etebook(param_ e model evaluate_the_m	, 'Adagra curacy': list): odel(trai (param)	[]}								t_value_in	nitial,efnB
accura toc = print accura Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35	acy_table = pd.DataF time.time() ("Minutes taken = " acy_table # display  1/4 8 [====================================	acy'].append( frame(accuracy + str((toc-ti the resullts	accuracy) _table) # c)/60.0))  [] - 1339s [] - 1317s [] - 1329s [] - 1334s	4s/step 4s/step 4s/step 4s/step	- loss: - loss: - loss:	2.5084 0.4882 0.3549 0.2783	- accuracy: - accuracy: - accuracy: - accuracy:	0.8281 · 0.8742 · 0.8973 ·	- val_loss	6: 0.2847 6: 0.2113	- val_a - val_a	ccuracy: 6	9.8815 9.9174
358/35 359/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch	8 [====================================		[ - 613s	2s/step - 4s/step 4s/step 4s/step 4s/step 2s/step - 5s/step 5s/step	- loss: - loss: - loss: - loss: - loss: - loss: - loss:	0.5367 - 0.8377 0.4519 0.3445 0.2578 0.2949 - 5.7865 0.7259	accuracy: 0 - accuracy: - accuracy: - accuracy: - accuracy: accuracy: 0 - accuracy: - accuracy:	0.8180 0.8180 0.8648 0.9036 0.8780 0.5975 0.7813	- val_loss - val_loss - val_loss - val_loss - val_loss - val_loss	6: 0.5529 6: 0.3014 6: 0.2320 6: 0.2949 6: 1.1571 6: 0.4371	<ul><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li></ul>	ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6	9.7780 9.8812 9.9080 9.8780 9.7087
Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch	3/4 8 [====================================		[] - 1619s [] - 1604s [] - 646s [] - 1401s [] - 1375s [] - 1354s [] - 1355s [] - 630s	5s/step  4s/step  4s/step  4s/step  4s/step  4s/step  4s/step  2s/step -	- loss:	0.6020 0.5373 0.2929 - 0.6773 0.3716 0.2576 0.1901 0.1756 -	- accuracy: - accuracy: accuracy: - accuracy: - accuracy: - accuracy: - accuracy: accuracy:	0.8340 0.8414 .8791 0.7233 0.8595 0.8959 0.9315 .9303	- val_loss - val_loss - val_loss - val_loss - val_loss - val_loss	6: 0.6526 6: 0.2929 6: 0.3400 6: 0.2172 6: 0.1697 6: 0.1756	<ul><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li></ul>	ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6	9.7819 9.8791 9.8746 9.9216 9.9324 9.9303
358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 359/35 Minute optin	8		[] - 1369s [] - 1363s [] - 1392s	4s/step 4s/step 4s/step	- loss: - loss:	0.4994 0.3879 0.3271	- accuracy: - accuracy: - accuracy:	0.8124 · 0.8732 · 0.8885 ·	- val_loss	6: 0.3684 6: 0.2866	- val_a - val_a	ccuracy: 6	9.8739 9.9084
2 RMS 3 Ada 4 Ada from p # plot rcPara ax = a ax.set	Sprop 87.909406  Agrad 93.031359  delta 92.926830  Dylab import rcParame  tresults  ams['figure.figsize'  accuracy_table.plot( t_xlabel("Optimizer"	] = 10,8 # wi x='optimizer' )			rle='bx-	', grid=	True)						
ax.set	t_ylabel("accuracy)" , 0.5, 'accuracy)')	•				-X- accu	racy						
91 — 90 accurack)													
# Get temp = optim: print print	optimum value for para accuracy_table[accuracy_table] accuracy = temp[para ("max Accuracy = %0.3 ("optimum" + paramcuracy = 93.031	Operaram curacy_table[' m_label].valu 3f" % accurac	es[0] y_table['	] == accu	].max()	ble[' <mark>acc</mark>	uracy'].max(	)]					
param param accuratic = for pa	ning dropol  _label = 'dropout_pro _list = [0.5, 0.6, 0  acy_table = {param_latime.time() aram_in_tqdm.tqdm_norm	ob' 0.7, 0.8, 0.9] abel: [], 'ac otebook(param_	curacy': list):										
accuratoc = print accura	Train, predict and occuracy, _ = train_excuracy, _ = train_excuracy_table[paramccuracy_table['accuracy_table = pd.DataFtime.time() ("Minutes taken = " accy_table # display  rs\Student\anaconda_	label].append acy'].append( rame(accuracy + str((toc-ti the resullts	odel(trai (param) accuracy) _table) # c)/60.0))	convert nsorflow\	the tab.	le to a d	dataframe gine\trainin	g.py:19					
will b warn Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch	e removed in a futur ings.warn('`Model.fi 1/4 8 [====================================	e version. Plat_generator`	ease use is deprec : 1345s : 1303s : 1302s : 1307s : 1240s	Model.fi ated and  4s/step  4s/step  4s/step  4s/step  2s/step -  3s/step	- loss: - loss: - loss: - loss: - loss: - loss:	3.2620 0.5616 0.3792 0.3954 0.3597 - 3.4531	- accuracy: - accuracy: - accuracy: - accuracy: - accuracy: accuracy: 0 - accuracy:	0.6565 0.7949 0.8627 0.8620 0.8864	- val_loss - val_loss - val_loss - val_loss	6: 0.5753 6: 0.3743 6: 0.2664 6: 0.3597 6: 0.5186	- val_a - val_a - val_a - val_a - val_a - val_a	ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6	9.8024 9.8585 9.8962 9.8864 9.8230
Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35	3/4 8		[] - 1217s [] - 1220s [] - 557s [] - 1353s [] - 1332s [] - 1340s [] - 1350s	3s/step 3s/step 2s/step 4s/step 4s/step 4s/step 4s/step	- loss: - loss: - loss: - loss: - loss: - loss:	0.4623 0.3925 0.2433 - 4.4965 0.7271 0.6005 0.4855	- accuracy: - accuracy: accuracy: - accuracy: - accuracy: - accuracy: - accuracy:	0.8417 - 0.8655 - 8962 - 0.5699 - 0.7320 - 0.7837 - 0.8319 - 0.8319	- val_loss - val_loss - val_loss - val_loss - val_loss	6: 0.3268 6: 0.2433 6: 0.7670 6: 0.4393 6: 0.4427	<ul><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li></ul>	ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6	9.8972 9.8962 9.6746 9.8268 9.8286
Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch 358/35 Epoch	8 [====================================		[] - 1376s [] - 1483s [] - 1553s [] - 721s [] - 1590s [] - 1509s	4s/step 4s/step 4s/step 2s/step 4s/step 4s/step	- loss: - loss: - loss: - loss: - loss:	0.7385 0.5969 0.4714 0.3470 - 9.7914 1.2089	- accuracy: - accuracy: - accuracy: accuracy: 0 - accuracy: - accuracy:	0.7638 - 0.8064 - 0.84498808 - 0.4071 - 0.4675 - 0.4675	- val_loss - val_loss - val_loss - val_loss - val_loss	6: 0.3943 6: 0.5374 6: 0.3470 6: 1.2320 6: 0.9424	<ul><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li></ul>	ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6	9.8516 9.7969 9.8808 9.5003 9.5557
359/35 Minute	8 [====================================	========							- val_loss	s: 0.9019	- val_a	ccuracy: 6	9.5777
# plot rcPara ax = a ax.set ax.set	oylab import rcParametresults ams['figure.figsize' accuracy_table.plot('accuracy)" t_ylabel("accuracy)" , 0.5, 'accuracy)')	] = 10,8 # wi x='dropout_pr			style='	bx-', gr:							
85 — 80 — 75 —													
65	0.50 0.55 0.60		0.70 0 out_prob	.75 0.8	80 C	0.85	0.90						
temp = dropou print print max Ac optimum	optimum value for parameter accuracy_table[accuracy= temp[parameter] ("max Accuracy = %0" ("optimum" + parameter accuracy = 89.617 m dropout_prob = 0.6 mg batch simple bat	uracy_table[' label].values 3f" % accurac label + " = "	[0] y_table[' + str(dr	accuracy' opout_opt	].max()	)		)]					
param_ param2 param2 accura tic = for pa	list = [1,2,5,10] #  2_label = 'batch_size 2_list = [8, 16] # [6  acy_table = {param_latime.time()} aram in tqdm.tqdm_nor br param2 in tqdm_nor  # generate train a train_generator,va # train, predict a	abel: [], par tebook(param_ tebook(param2 and validational_generator, and evaluate	using the am2_label list): _list): n data _ = get_ model	: [], ' <mark>a</mark> c data_gene	curacy'	: []}	aset_path,va	lidatior	n_dataset_	_path,test	t_datase	t_path, ta	argetsize_c
toc = print(accura	<pre># collect results accuracy_table[pa accuracy_table[pa accuracy_table['accuracy_table]'accuracy_table accuracy_table = pd.DataF time.time() ("Minutes taken = " accy_table # display</pre> on-input-13-222ed610	ram_label].ap ram2_label].a ccuracy'].app rame(accuracy + str((toc-ti the resullts	pend(para ppend(para end(accura _table) # c)/60.0))	m) am2) acy) <i>convert</i> ionWarnir	the tab.	le to a (	dataframe				out_valu	e_initial,	efnB7_bas
Found Found Found C:\Use will b warn 358/35 359/35	use `tqdm.notebook. param2 in tqdm_noteb  2870 images belongin  2870 images belongin  394 images belonging rs\Student\anaconda_ e removed in a futur ings.warn('`Model.fi  8 [===================================	oook(param2_ling to 4 classes to 4 classes alib\site-pare version. Plat_generator alignments of the classes to 4 classes to 4 classes to 4 classes alignments of the classes a	st): s. ckages\te ease use is deprec ] - 1394s ] - 620s s. s 1290s	nsorflow\ `Model.fi ated and 4s/step 2s/step -	<pre>Apython\ it`, whi     loss:     loss:     loss:</pre>	3.3703 0.5212 -	rts generato - accuracy: accuracy: 0 - accuracy:	0.6499 0.8254	- val_loss	s: 0.5212	- val_a	ccuracy: 6	0.8254
Found Found 179/17	2870 images belongin 2870 images belongin 2870 images belongin 394 images belonging 1/2 8 [====================================	ng to 4 classes g to 4 classes	5] - 598s 5. 6] - 1371s 7] - 1367s 8] - 628s 85.	3s/step - 4s/step 4s/step 2s/step -	- loss: - loss: - loss: - loss:	3.2866 0.5681 0.3039 -	accuracy: 0 - accuracy: - accuracy: accuracy: 0	0.6408 - 0.7883 - .8937	- val_loss - val_loss	6: 0.5142 6: 0.3039	- val_a - val_a	ccuracy: 6	9.8007 9.8937
Found Found 179/17 180/18 Found Found Epoch 358/35 Epoch 358/35 Found Found Found Found Found Found	9 [====================================	ng to 4 classeng to 4 classes	[] - 1265s [] - 602s [] - 602s [] - 1420s [] - 1388s [] - 1383s	7s/step 3s/step - 4s/step 4s/step 4s/step	- loss: - loss: - loss: - loss:	0.5198 0.2805 - 3.1804 0.5633 0.3921	- accuracy: accuracy: - accuracy: - accuracy: - accuracy:	0.80668948	- val_loss - val_loss - val_loss	6: 0.4951 6: 0.3516 6: 0.2836	<ul><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li></ul>	ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6	9.8948 9.8125 9.8746 9.8913
Found	4/5		- :] - 1384s :] - 1379s	4s/step	- loss: - loss: - loss:	0.3123 0.3163 0.1391 - 3.9963 0.5333 0.4186	- accuracy: - accuracy: accuracy: 0	0.8871 - 0.8973 - 9554 - 0.6440 - 0.7933 - 0.8525 - 0.8525	- val_loss - val_loss - val_loss - val_loss - val_loss	6: 0.1842 6: 0.1391 6: 0.4225 6: 0.3224 6: 0.2204	<ul><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li></ul>	ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6 ccuracy: 6	9.9286 9.9554 9.8341 9.8843 9.9240
Found	4/5 8 [====================================	to 4 classes		7s/step 7s/step	- loss:	P 0400	-	0.9075			- val_a	ccuracy: 6 ccuracy: 6 ccuracy: 6	9.9491 9.7265 9.8662
Found	4/5 8 [====================================	g to 4 classes	[] - 1248s [] - 1240s [] - 1251s [] - 1255s [] - 1257s [] - 600s [] - 1431s [] - 1425s	7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step	- loss: - loss: - loss: - loss: - loss:	0.2574 0.1247 - 3.3611 0.6329	- accuracy: - accuracy: 0 - accuracy: - accuracy:	0.7771 -	- val_loss	s: 0.3551	- **	ccuracy: 6	9.9118 9.9105 9.9449 9.9474 9.9108 9.9443
Found	4/5 8 [====================================	ng to 4 classes  ang to 4 classes	1248s   1240s   1251s   1255s   1257s   1257s   1257s   1400s   1404s   1404s   1406s   1406s   1406s   1406s   1406s   1405s   1405	7s/step 7s/step 7s/step 7s/step 3s/step 4s/step	- loss:	0.2574 0.1247 - 3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035	- accuracy:	0.7771 · 0.8421 · 0.8679 · 0.8847 · 0.9039 · 0.9214 · 0.8980 · 0.9168 · 0.9333 · · 0.9333 · · 0.9333 · · 0.9333 · · 0.9333 · · 0.9333 · · 0.9333 · · 0.9333 · · 0.9333 · · 0.9333 · · 0.9333 · · 0.9333 · · · 0.9333 · · · 0.9333 · · · · · · · · · · · · · · · · · ·	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153	<ul><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li><li>val_a</li></ul>	ccuracy: 6	
Found Found 179/17 180/18  Found Fou	4/5 8 [====================================	ng to 4 classes  The second of	1248s   1240s   1251s   1255s   1257s   1257s   1404s   1405s   1406s   1405s   1407s   1407s   1406s   1407s   1407	7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step 7s/step 7s/step 7s/step 7s/step 7s/step 7s/step	- loss:	0.2574 0.1247 - 3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 - 4.3885 0.5320 0.4459 0.3292 0.2884	- accuracy:	0.7771 - 0.8421 - 0.8679 - 0.8847 - 0.9039 - 0.9214 - 0.8980 - 0.9168 - 0.9333 - 9505 - 0.6345 - 0.8069 - 0.8441 - 0.8819 - 0.8966 - 0.8966 - 0.8966 - 0.8966 - 0.8966 - 0.8966 - 0.8966 - 0.8966 - 0.8966 - 0.8966 - 0.8966	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153 6: 0.2153 6: 0.2280 6: 0.2280 6: 0.2556 6: 0.1549	- val_a	ccuracy: 6	9.8951 9.9143 9.9073 9.9488
Found 179/17 180/18 Found Foun	4/5 8 [====================================	ng to 4 classes  ang to 4 classes	1248s   1240s   1251s   1255s   1257s   1257s   1404s   1404s   1404s   1406s   1405s   1405s   1405s   1405s   1405s   1406s   1406	7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss:	0.2574 0.1247 - 3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 - 4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1806 0.1912	- accuracy:	0.7771 - 0.8421 - 0.8679 - 0.8847 - 0.9039 - 0.9214 - 0.8980 - 0.9168 - 0.9333 - 9505 - 0.6345 - 0.8069 - 0.8441 - 0.8819 - 0.8819 - 0.8966 - 0.9236 - 0.9236 - 0.9236 - 0.9236 - 0.9320 - 0.9425 - 0.9366 - 0.937	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153 6: 0.4073 6: 0.4073 6: 0.2999 6: 0.2280 6: 0.2556 6: 0.1549 6: 0.1549 6: 0.1906 6: 0.1484 6: 0.1181 6: 0.2464	- val_a	ccuracy: 6	9.8951 9.9143 9.9073 9.9488 9.9272 9.9425 9.9578
Found 179/17 180/18   Found Found 179/17 180/18   Found Foun	4/5 88 [===================================	g to 4 classes	1248s   1240s   1251s   1255s   1257s   1257s   1404s   1404s   1404s   1406s   1405s   1405s   1405s   1405s   1405s   1406s   1406	7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss:	0.2574 0.1247 - 3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 - 4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1806 0.1912	- accuracy:	0.7771 - 0.8421 - 0.8679 - 0.8847 - 0.9039 - 0.9214 - 0.8980 - 0.9168 - 0.9333 - 9505 - 0.6345 - 0.8069 - 0.8441 - 0.8819 - 0.8819 - 0.8966 - 0.9236 - 0.9236 - 0.9236 - 0.9236 - 0.9320 - 0.9425 - 0.9366 - 0.937	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153 6: 0.4073 6: 0.4073 6: 0.2999 6: 0.2280 6: 0.2556 6: 0.1549 6: 0.1549 6: 0.1906 6: 0.1484 6: 0.1181 6: 0.2464	- val_a	ccuracy: 6	9.8951 9.9143 9.9073 9.9488 9.9272 9.9425 9.9578
Found 179/17 180/18   Found Found 179/17 180/18   Found Foun	### ### ### ### ### ### ### ### ### ##	g to 4 classes  g to 5 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9 classes	1248s   1240s   1251s   1255s   1257s   1257s   1404s   1405s   1406s   1406s   1406s   1406s   1406s   1406s   1406s   1405s   1407s   1264s   1265s   1265	7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss:	0.2574 0.1247 -  3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1806 0.1912 0.0814 -	- accuracy:	0.7771 - 0.8421 - 0.8679 - 0.8847 - 0.9039 - 0.9168 - 0.9333 - 0.9505 - 0.6345 - 0.8069 - 0.8441 - 0.8819 - 0.8966 - 0.9236 - 0.9320 - 0.9425 - 0.9366 - 0.9376 - 0.9777	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153 6: 0.4073 6: 0.2999 6: 0.2280 6: 0.2556 6: 0.1549 6: 0.1906 6: 0.1484 6: 0.1181 6: 0.2464 6: 0.0814	- val_a	ccuracy: 6	9.8951 9.9143 9.9073 9.9488 9.9272 9.9425 9.9578
Found 179/17 180/18   Found Fo	#/5 # [====================================	g to 4 classes  g to 5 classes  g to 6 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9 classes	idth 10, aram2_strict	7s/step 7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss:	0.2574 0.1247 -  3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1960 0.1685 0.1806 0.1912 0.0814 -	- accuracy:	0.7771 - 0.8421 - 0.8679 - 0.8847 - 0.9039 - 0.9168 - 0.9333 - 0.9505 - 0.8441 - 0.8819 - 0.8866 - 0.9236 - 0.9320 - 0.9425 - 0.9366 - 0.9777 - 0.9777	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153 6: 0.4073 6: 0.2999 6: 0.2280 6: 0.2556 6: 0.1549 6: 0.1906 6: 0.1484 6: 0.1181 6: 0.0814	- val_a	ccuracy: 6	9.8951 9.9143 9.9073 9.9488 9.9272 9.9425 9.9578
Found 179/17 180/18   Found Found 179/17 180/18   Found Foun	# # # # # # # # # # # # # # # # # # #	g to 4 classes  g to 5 classes  g to 6 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9 classes	idth 10, aram2_strict	7s/step 7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss:	0.2574 0.1247 -  3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1960 0.1685 0.1806 0.1912 0.0814 -	- accuracy: - accu	0.7771 - 0.8421 - 0.8679 - 0.8847 - 0.9039 - 0.9168 - 0.9333 - 0.9505 - 0.8441 - 0.8819 - 0.8866 - 0.9236 - 0.9320 - 0.9425 - 0.9366 - 0.9777 - 0.9777	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153 6: 0.4073 6: 0.2999 6: 0.2280 6: 0.2556 6: 0.1549 6: 0.1906 6: 0.1484 6: 0.1181 6: 0.0814	- val_a	ccuracy: 6	9.8951 9.9143 9.9073 9.9488 9.9272 9.9425 9.9578
Found	4/5 8 [====================================	g to 4 classes  g to 5 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9 classes	s. s	7s/step 7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss:	0.2574 0.1247 -  3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1960 0.1685 0.1806 0.1912 0.0814 -	- accuracy: - accu	0.7771 - 0.8421 - 0.8421 - 0.8679 - 0.8847 - 0.9039 - 0.9214 - 0.8980 - 0.9333 - 0.9505 - 0.8441 - 0.8819 - 0.8866 - 0.9236 - 0.9	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153 6: 0.4073 6: 0.2999 6: 0.2280 6: 0.2556 6: 0.1549 6: 0.1906 6: 0.1484 6: 0.1181 6: 0.0814	- val_a	ccuracy: 6	9.8951 9.9143 9.9073 9.9488 9.9272 9.9425 9.9578
Found	# # # # # # # # # # # # # # # # # # #	g to 4 classes  g to 5 classes  g to 6 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9 classes	idth 10, aram2_labacy', sty.  idth 10, aram2_labacy', sty.  idth 10, aram2_labacy', sty. idth 10, aram3_labacy', sty. idth 10, aram3_labacy', sty. idth 10, aram3	7s/step 7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss: - loss	0.2574 0.1247 -  3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1806 0.1912 0.0814 -  [0]] rue)  [0]] rue)  [0]] rue)	- accuracy: - accu	0.7771 - 0.8421 - 0.8421 - 0.8679 - 0.9214 - 0.8980 - 0.9333 - 0.9505 - 0.6345 - 0.8819 - 0.8819 - 0.8966 - 0.9366 - 0.9376 - 0.9	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153 6: 0.4073 6: 0.2999 6: 0.2280 6: 0.2556 6: 0.1549 6: 0.1906 6: 0.1484 6: 0.1181 6: 0.0814	- val_a	ccuracy: 6	9.8951 9.9143 9.9073 9.9488 9.9272 9.9425 9.9578
Found	### ### ### ### ### ### ### ### ### ##	ptimum  g to 4 classes  and to 4 classes  g to 4 classes  and to 4 classes  g to 5 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9 classes  g to 9 classes  g to 9 classes  g to 10 classes  g to 10 classes  g to 2 classes  g to 3 classes  g to 4 classes  g to 4 classes  g to 4 classes  g to 4 classes  g to 5 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9 classes	idth 10, aram2	7s/step 7s/step 7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss: - loss	0.2574 0.1247 -  3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1806 0.1912 0.0814 -  ble['acc' )  0.5)) #  ble['ixlen(')  0.5)) #	- accuracy: - accu	0.7771 - 0.8421 - 0.8679 - 0.8847 - 0.9039 - 0.9214 - 0.8980 - 0.9333 - 0.9505 - 0.6345 - 0.8869 - 0.8441 - 0.8819 - 0.8966 - 0.9376 - 0.9	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153 6: 0.4073 6: 0.2999 6: 0.2280 6: 0.2556 6: 0.1549 6: 0.1906 6: 0.1484 6: 0.1181 6: 0.0814	- val_a	ccuracy: 6	9.8951 9.9143 9.9073 9.9488 9.9272 9.9425 9.9578
Found	## Staken = 883.735692	ptimum    continue   c	1248s   1240s   1251s   1255s   1257s   1255s   1257s   1404s   1405s   1406s   1406s   1406s   1406s   1406s   1407s   1406s   1407s   1407	7s/step 7s/step 7s/step 7s/step 7s/step 7s/step 7s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss: - loss	0.2574 0.1247 -  3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1806 0.1912 0.0814 -  [0]] rue)  list[i]] t[i%len(i) 0.5)) #	- accuracy: - accu	0.7771 · 0.8421 · 0.8679 · 0.8847 · 0.9039 · 0.9214 · 0.8980 · 0.9333 · 0.9168 · 0.9333 · 0.9425 · 0.9366 · 0.9376 · 0.9376 · 0.9377 ·   batch_si  batch_si  e_initia  e_initia  e_initia	- val_loss	6: 0.3551 6: 0.4095 6: 0.2233 6: 0.2563 6: 0.1500 6: 0.1583 6: 0.2701 6: 0.2153 6: 0.4073 6: 0.2999 6: 0.2280 6: 0.2556 6: 0.1549 6: 0.1906 6: 0.1484 6: 0.1181 6: 0.0814	- val_a	ccuracy: 6	9.8951 9.9143 9.9073 9.9488 9.9272 9.9425 9.9578
Found	## ## ## ## ## ## ## ## ## ## ## ## ##	ptinum  g to 4 classes  g to 5 classes  g to 6 classes  g to 6 classes  g to 6 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9	s. s	7s/step 7s/step 7s/step 7s/step 7s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss: - loss	0.2574 0.1247 -  3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1912 0.1806 0.1912 0.0814 -   (O]] rue)  hist[i]] t[i%len(i) 0.5)) #	- accuracy: - accu	0.7771 · 0.8421 · 0.8421 · 0.8679 · 0.8847 · 0.9039 · 0.9214 · 0.8980 · 0.9333 · 0.9505 · 0.8441 · 0.8819 · 0.8866 · 0.9236 · 0.9320 · 0.9425 · 0.9366 · 0.9376 · 0.9777 · d.	- val_loss	a: 0.3551 a: 0.4095 a: 0.2233 a: 0.2563 a: 0.1500 a: 0.1583 a: 0.2701 a: 0.2153 a: 0.2153 a: 0.2999 a: 0.2280 a: 0.2556 a: 0.1549 a: 0.1844 a: 0.1844 a: 0.1844 a: 0.1844 a: 0.0814	- val_a	ccuracy: 6	0.8951 0.9143 0.9073 0.9488 0.9272 0.9425 0.9578 0.9233 0.9777
Found	## A ST	ptinum  g to 4 classes  g to 5 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 8 classes  g to 9	1248s   1240s   1257s   1257s   1257s   1257s   1260s   1405s   1405s   1406s   1405s   1406s   1405s   1406s   1405s   1406s   1405s   1406s   1405s   1406s   1405s   1405	7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step 8s/step	- loss:	0.2574 0.1247 -  3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1685 0.1806 0.1912 0.1685 0.1806 0.1912 0.0814 -  (O]  (O)  (O)  (O)  (O)  (O)  (O)  (O)	- accuracy:	0.7771 : 0.8421 : 0.8421 : 0.8679 : 0.9214 : 0.9333 : 0.9505 : 0.9333 : 0.9505 : 0.8441 : 0.8819 : 0.8842 : 0.9326 : 0.9	- val_loss	arning: \footnote{\footnot	- val_a	path, test_ ccuracy: 6	efnB7_bas  efnB7_bas  or is depr  0.8010 0.8777  0.8010 0.8746 0.9038 0.9481 0.9495 0.9690 0.9547
Found	## ## ## ## ## ## ## ## ## ## ## ## ##	ptimum  g to 4 classes  g to 5 classes  g to 6 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9 classes  g to 9 classes  g to 9 classes  g to 1 classes  g to 1 classes  g to 2 classes  g to 3 classes  g to 4 classes  g to 4 classes  g to 4 classes  g to 5 classes  g to 6 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9	1248s	7s/step 7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step 8s/step	- loss:	0.2574 0.1247 -  3.3611 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.2260 0.1960 0.1685 0.1806 0.1912 0.0814 -  (0]] rue)  1.5t[i]] t[i%len() 0.5)) #  0.5)) #	- accuracy: - accu	batch_si	- val_loss	arning: \footnote{\text{3}} \text{3} : 0.3551 \text{3} : 0.4095 \text{3} : 0.2563 \text{3} : 0.2563 \text{3} : 0.1583 \text{3} : 0.2701 \text{3} : 0.2153 \text{3} : 0.2999 \text{3} : 0.2999 \text{3} : 0.2999 \text{3} : 0.2556 \text{3} : 0.1514 \text{3} : 0.1906 \text{3} : 0.1906 \text{3} : 0.1906 \text{3} : 0.1814 \text{3} : 0.1814 \text{3} : 0.0814	- val_a	path, test_ ccuracy: 6	efnB7_bas efnB7_bas or is depr 0.8910 0.8910 0.8910 0.8910 0.8943 0.9938 0.9947 0.9697 0.9697 0.9843
Found	## A	ptinum  g to 4 classes  g to 5 classes  g to 6 classes  g to 6 classes  g to 6 classes  g to 7 classes  g to 8 classes  g to 9	1248s   1248s   1240s   1257s   1257s   1257s   1260s   125s   1260s	7s/step 7s/step 7s/step 7s/step 7s/step 7s/step 3s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 4s/step 7s/step	- loss: - loss	0.2574 0.1247 - 0.1247 - 0.1247 - 0.1247 - 0.6329 0.4571 0.4033 0.3438 0.2772 0.2436 0.3528 0.3035 0.2224 0.1514 -  4.3885 0.5320 0.4459 0.3292 0.2884 0.0260 0.1960 0.1685 0.1806 0.1912 0.1806 0.1912 0.0814 -  0.1912 0.188 0.1602 0.1960 0.5183 0.2588 0.2343 0.1878 0.1602 0.1960 0.5183 0.3949 0.3253 0.1878 0.1602 0.1960 0.5183 0.2588 0.2343 0.1878 0.1602 0.1960 0.5183 0.2588 0.2343 0.1878 0.1602 0.1960 0.1806 0.1912 0.1880 0.2588 0.2343 0.1878 0.1602 0.1960 0.1802 0.1802 0.1802 0.1802 0.1802 0.1802 0.1802 0.1802 0.1802 0.1802 0.1960 0.1	- accuracy: - accu	## Company	- val_loss	arning: \footnote{\text{i}} \text{ar o. 3551} \text{ar o. 4095} \text{ar o. 2233} \text{ar o. 2563} \text{ar o. 1500} \text{ar o. 1583} \text{ar o. 1514} \text{ar o. 2701} \text{ar o. 2999} \text{ar o. 2999} \text{ar o. 2999} \text{ar o. 2999} \text{ar o. 1549} \text{ar o. 1549} \text{ar o. 1484} \text{ar o. 1814} \text{ar o. 1240} \text{ar o. 1814} \t	dataset_ val_a	ccuracy: 6	0.8951 0.9143 0.9073 0.9488 0.9272 0.9425 0.9578 0.9233 0.9777 0.8010 0.8746 0.9038 0.9481 0.9495 0.9690 0.9547 0.9697 0.9843 0.9819