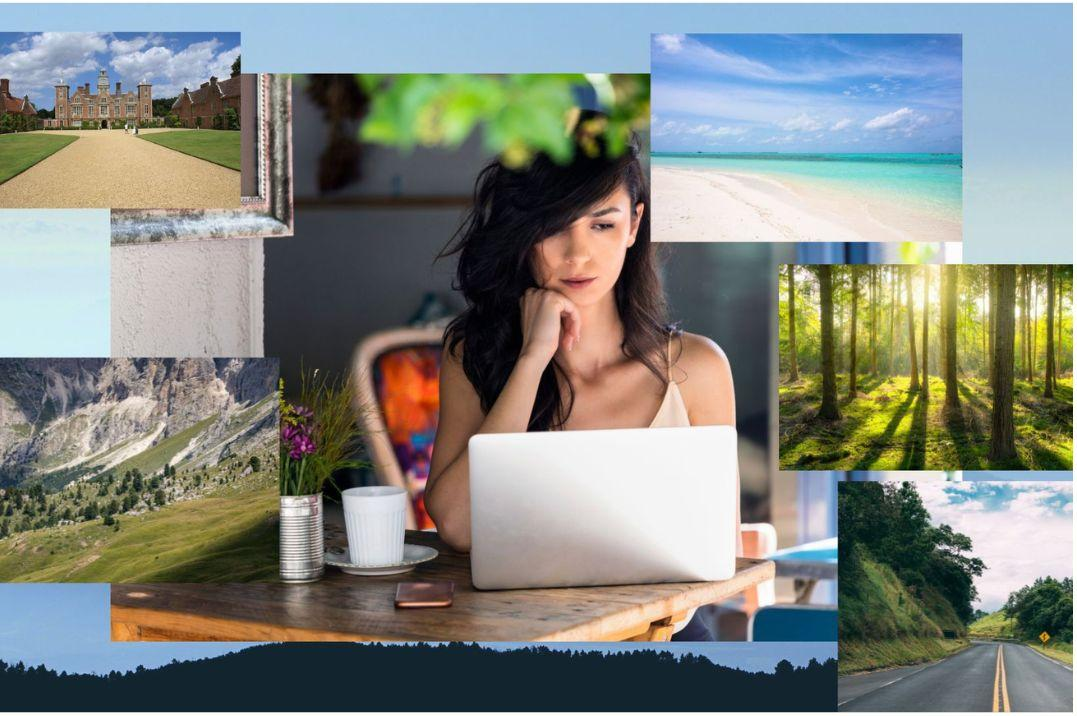
SCENE CLASSIFICATION

DATASCI 207 FINAL PROJECT OF TEAM : KT NORTON, YURI KINAKIN & JUDD GALLARES
PROF. NEDELINA TENEVA, PHD

DECEMBER 6, 2022



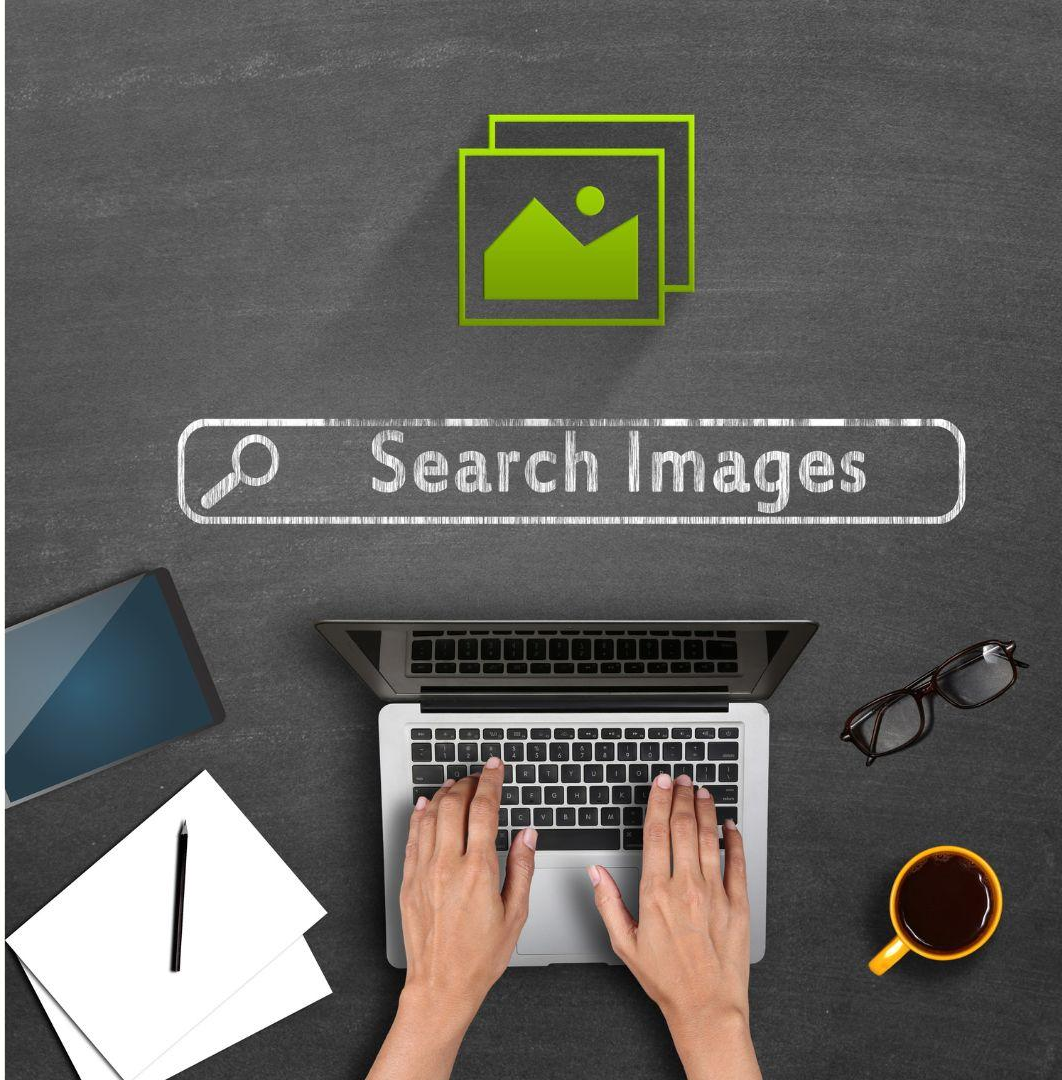
Problem Motivation

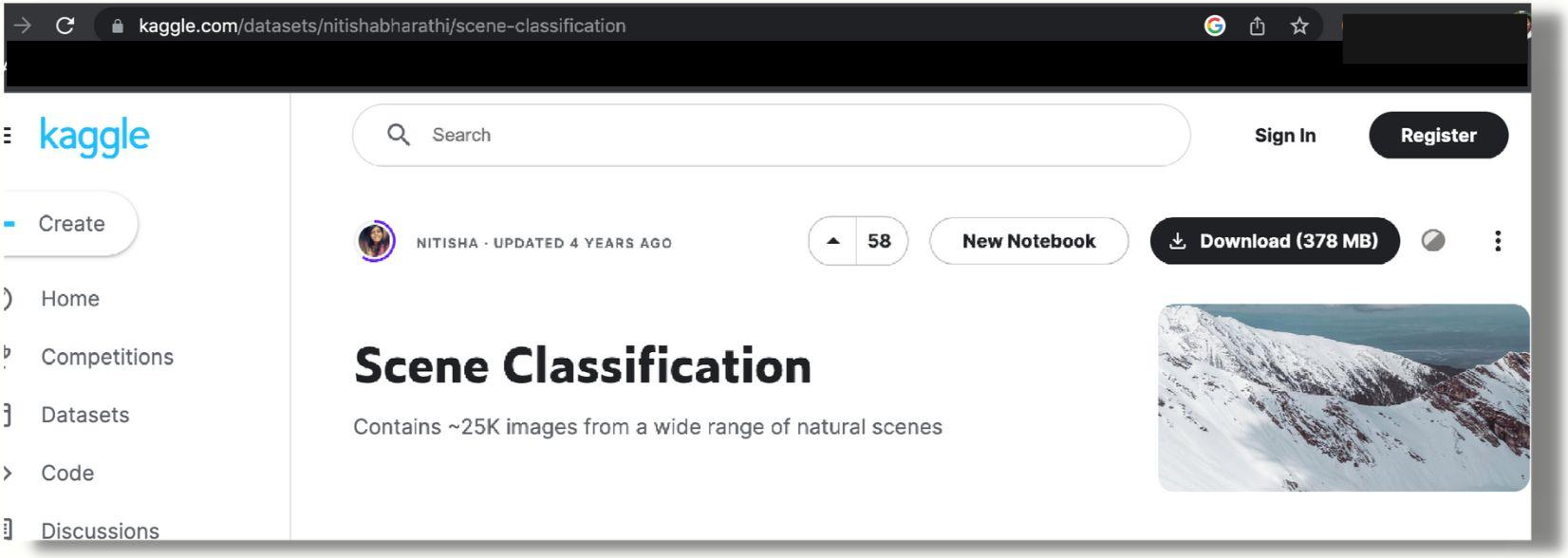
We work for a photography travel blog website that allows its users to share, view and interact with other users' content and photography.

Berkeley
UNIVERSITY OF CALIFORNIA

Problem Motivation

We want to recommend blog articles to our users based on photos that they have liked, interacted with, or posted





Data source: Kaggle.com

Problem Statement

We will explore various techniques to classify images based on their natural scene or setting.

Berkeley
UNIVERSITY OF CALIFORNIA

Dataset Description

The data contains ~25k images from around the world of natural outdoor scenes, classified in one of 6 classes.

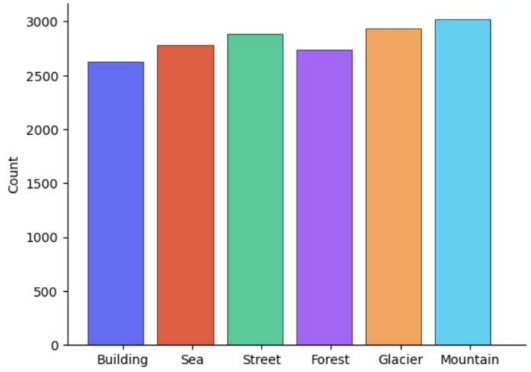
1. FORESTS
2. MOUNTAINS
3. STREETS
4. SEA
5. BUILDINGS
6. GLACIERS

Berkeley
UNIVERSITY OF CALIFORNIA

Dataset Description

train	A folder containing 24,335 of mostly 150x150px colored (rgb) jpegs	 - 1.jpg - 12.jpg - 123.jpg ...
train.csv	A csv file with 17,034 rows of labels each describing an image in **train**	 - Buildings - Forests - Mountains - Glaciers - Street - Sea
test_WyRytb0.csv	A csv file containing 7,301 rows of image names (strings) corresponding to a jpeg in **train**. The **train** and **test** csvs should have no duplicates.	 - “3.jpg” - “10.jpg” - “168.jpg” ...

Exploratory Data Analysis



Dataset Summary

A total of 16,979 labelled images (150px x 150px x 3rgb channels) are available for training.

They are split into 6 different categories of approximately equal size.

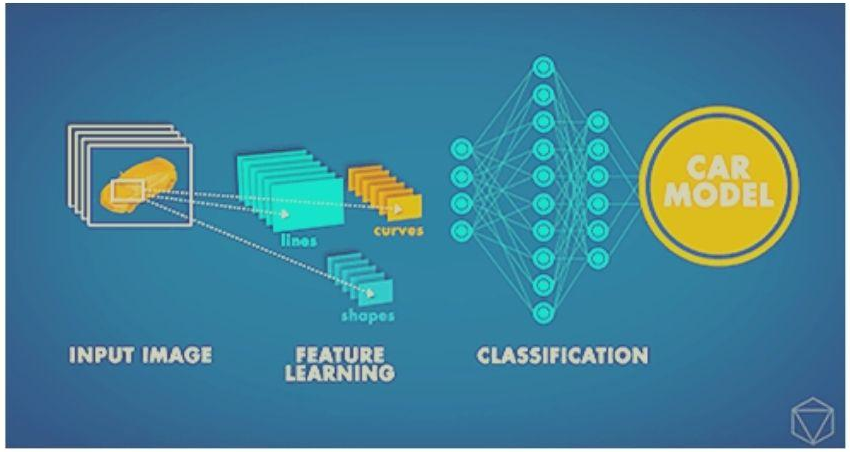
Our baseline accuracy for each class is therefore **~17%.**

Berkeley
UNIVERSITY OF CALIFORNIA

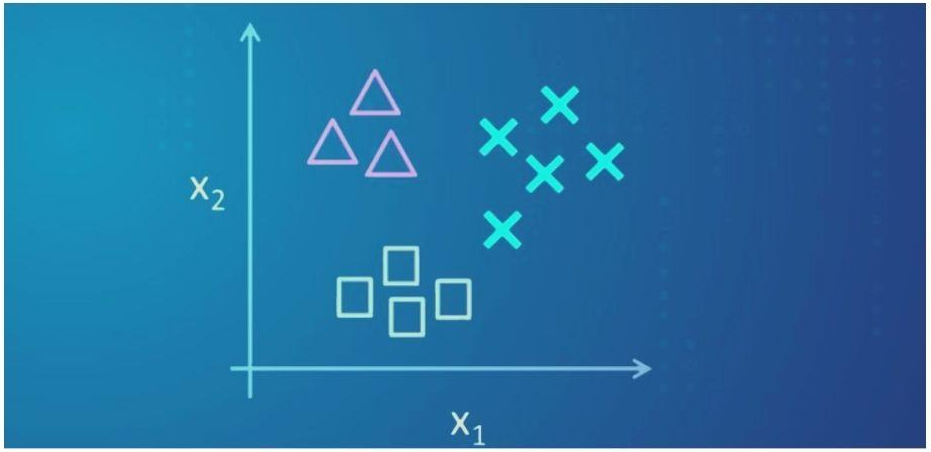
Solution/Approach Description

Our initial analysis utilizing classical machine learning techniques result in a relatively low and imbalance classification accuracy. Our team then tested the approaches below to see which one produced the best results.

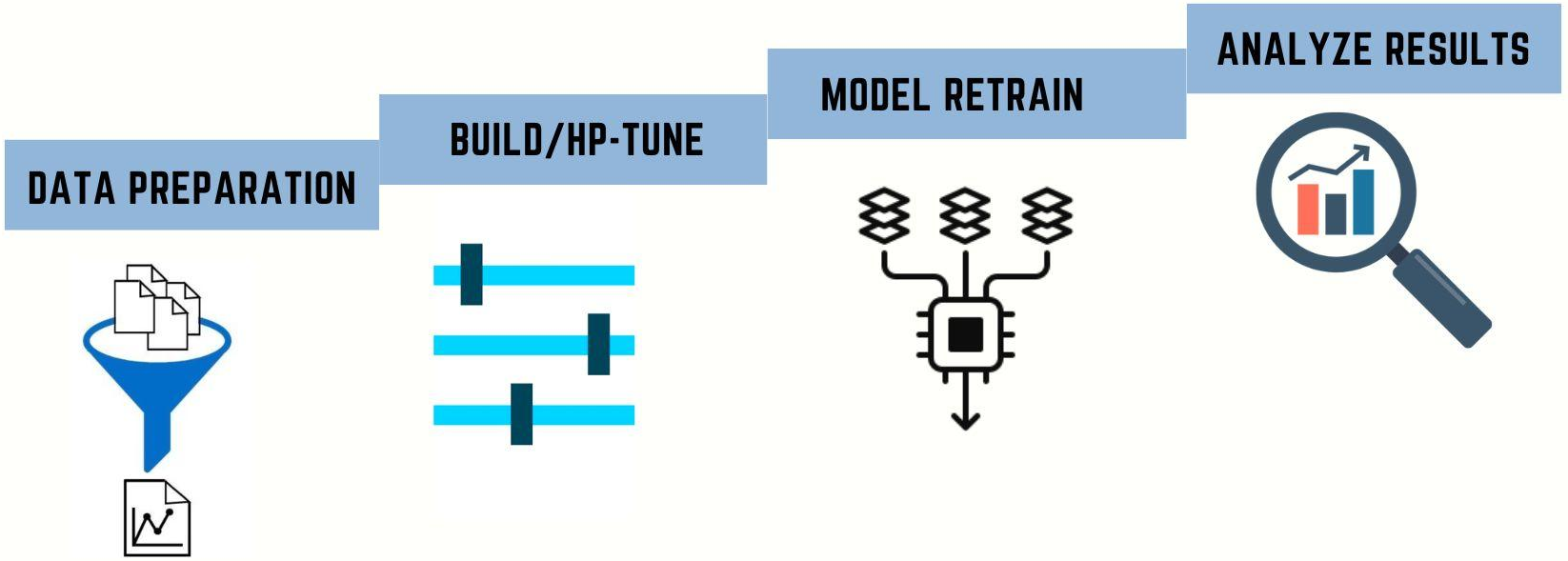
Convolutional Neural Network (CNN) 2D - RGB and 1D - Grayscale



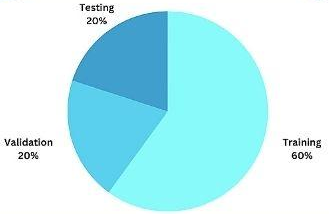
Multiclass Logistic Regression - Grayscale



Model Experiment Process

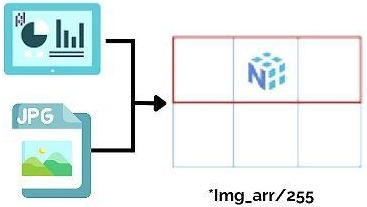


2. Generate an equal number of images from each class



4. Split the images to comprise 60% training, 20% validation and 20% testing

1. Choose only images of size 150x150x3



3. Convert list of labels and images to numpy arrays

5. Shuffle the images and labels together.

Data Preparation and Transformation

Berkeley
UNIVERSITY OF CALIFORNIA

Convolutional Neural Network 2D

Input 150x150 pixels x 3 (RGB)

Preprocessing

Convolution 2D

16 layers + ReLU + 3x3 kernel size

Subsampling

Max pooling 2x2 size

FLATTEN

BUILDINGS

FOREST

SEAS

MOUNTAINS

STREETS

GLACIERS

OUTPUT LAYER

FEATURE EXTRACTION

SOFTMAX CLASSIFICATION

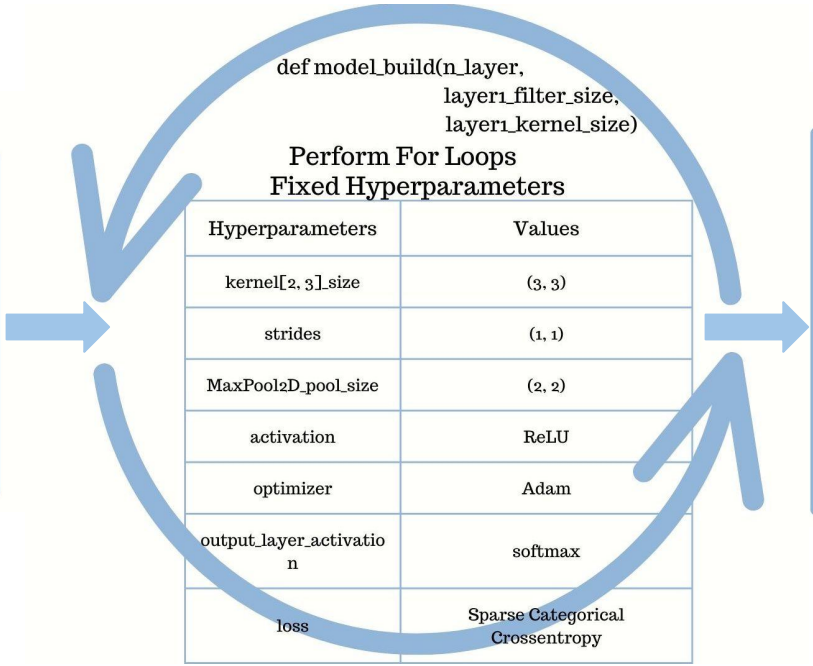
Berkeley
UNIVERSITY OF CALIFORNIA

Hyperparameter Tuning

INPUT

Hyperparameters to vary

Hyperparameters	Values
n_layers	1, 2, 3
layer1_filter_size	64, 32, 16
layer1_kernel size	(5, 5), (3, 3), (1, 1)



Hyperparameters	Values
kernel[2, 3]_size	(3, 3)
strides	(1, 1)
MaxPool2D_pool_size	(2, 2)
activation	ReLU
optimizer	Adam
output_layer_activation	softmax
loss	Sparse Categorical Crossentropy

OUTPUT

Cases	loss	accu	val_loss	val_accu
[1,64,(5,5)]	0.0xxxa1	0.0xxxa2	0.0xxxa3	0.0xxxa4
[1,64,(3,3)]	1.0xxxb1			
[1,64,(1,1)]	2.0xxxc1			
[1,32,(5,5)]	0.0xxxd1			
...	...	...	...	...
[3,16,(1,1)]	0.0xxxn1			

Berkeley
UNIVERSITY OF CALIFORNIA

Model Retrain: Apply CNN_2D Optimum Case

	train_loss	train_accu	val_loss	val_accu	filter1_size	kernel1_size	num_conv_layers	build_minutes
0	0.0031	0.9999	0.0361	0.9905	64.0	(5, 5)	1	8.879024
1	0.0058	0.9996	0.0347	0.9911	32.0	(5, 5)	1	4.789623
2	0.0077	0.9997	0.0368	0.9895	16.0	(5, 5)	1	3.473544
3	0.0033	0.9998	0.0263	0.9930	64.0	(3, 3)	1	6.649201
4	0.0040	0.9998	0.0288	0.9924	32.0	(3, 3)	1	4.994755
5	0.0081	0.9998	0.0299	0.9933	16.0	(3, 3)	1	2.412871
6	0.0824	0.9823	0.1419	0.9705	64.0	(1, 1)	1	6.837631
7	0.1238	0.9722	0.1747	0.9559	32.0	(1, 1)	1	3.754270
8	0.1947	0.9533	0.2762	0.9194	16.0	(1, 1)	1	2.064861
9	0.0596	0.9861	0.0870	0.9832	64.0	(5, 5)	2	13.014805

CNN_2D *Model Retrain* at optimum HPs over more epochs

HYPERPARAMETERS	VALUES
Epochs	**10** [5, 10, 15]
No. of CNN Layers	**1** [1, 2, 3]
Filter Size	**64** [64, 32, 16]
Kernel Size	**(3, 3)** [(5, 5), (3, 3), (1, 1)]
MaxPool2D Size	*(2, 2)*
Conv2D activation	*ReLU*
Output Activation	*Softmax*
Optimizer	*Adam*
Loss	*SCC*

	loss	accuracy	val_loss	val_accuracy
0	1.242730	0.703500	0.318093	0.920368
1	0.130304	0.969546	0.085889	0.981599
2	0.026977	0.996511	0.052284	0.991434
3	0.008711	0.999471	0.027462	0.992703
4	0.003313	0.999789	0.026319	0.993020
5	0.001790	1.000000	0.036934	0.991434
6	0.001181	1.000000	0.026425	0.993338
7	0.000731	1.000000	0.026440	0.993655
8	0.000528	1.000000	0.026746	0.993338
9	0.000399	1.000000	0.025375	0.995241

Analyze Results: CNN_2D Classification Report

There are no indication of bias in the classes, 0 - 5. They are tightly grouped together between > 0.98 and < 1 in all metrics.

Classes

0 - Buildings
1 - Forests
2 - Glacier
3 - Mountains
4 - Sea
5 - Street

	precision	recall	f1-score	support
0	0.9978	0.9803	0.9890	458.0000
1	1.0000	1.0000	1.0000	508.0000
2	0.9944	0.9832	0.9888	537.0000
3	0.9852	0.9962	0.9907	533.0000
4	0.9894	1.0000	0.9947	466.0000
5	0.9908	0.9954	0.9931	651.0000
accuracy	0.9927	0.9927	0.9927	0.9927
macro avg	0.9929	0.9925	0.9927	3153.0000
weighted avg	0.9927	0.9927	0.9927	3153.0000

Precision - Percentage of correct positive predictions relative to total positive predictions.

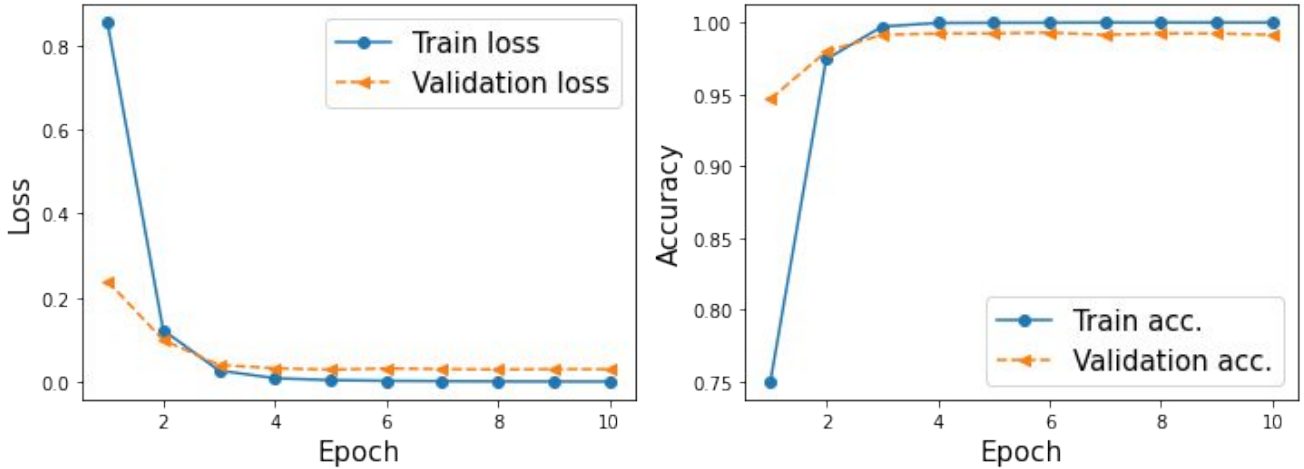
Recall - Percentage of correct positive predictions relative to total actual positives.

F1-score - A weighted harmonic mean of precision and recall: 2 * (Precision * Recall) / (Precision + Recall)

Support - These values tell us how many images belong to each class in the test dataset.

Berkeley
UNIVERSITY OF CALIFORNIA

Analyze Results: CNN_2D Loss and Accuracy Plots



There are no indications of over-fitting as can be seen above where the plots converged almost from the start.

Convolutional Neural Network 1D

Input 150x150 pixels x
1 Grayscale

Preprocessing

Convolution 1D

256 layers + ReLU + kernel size = 1

Subsampling

Max pooling size = 2

F
L
A
T
T
E
N

BUILDINGS
FORESTS
SEAS
MOUNTAINS
STREETS
GLACIERS

OUTPUT LAYER

FEATURE EXTRACTION

SOFTMAX CLASSIFICATION

Berkeley
UNIVERSITY OF CALIFORNIA

CNN_1D: Hypertune and Apply Optimum Case

	train_loss	train_accu	val_loss	val_accu	filter1_size	kernel1_size	build_minutes
0	1.0878	0.6101	1.0513	0.6164	512.0	7	1.086736
1	1.0349	0.6254	0.9917	0.6304	256.0	7	0.785027
2	1.1566	0.5726	1.1152	0.6028	128.0	7	0.524590
3	1.0901	0.5986	1.0910	0.5945	64.0	7	0.366595
12	0.1582	0.9662	0.1822	0.9578	512.0	1	0.568469
13	0.1595	0.9712	0.1890	0.9626	256.0	1	0.362797
14	0.2300	0.9508	0.2843	0.9334	128.0	1	0.233020
15	0.3917	0.9005	0.4112	0.8877	64.0	1	0.214232

Berkeley
UNIVERSITY OF CALIFORNIA

CNN_1D *Model Retrain* at optimum HPs over more epochs

HYPERPARAMETERS	VALUES
Epochs	**15** [5, 10, 15, 20]
No. of CNN Layers	**1** [1, 2, 3]
Filter Size	**256** [512, 256, 128, 64]
Kernel Size	**1** [7, 5, 3, 1]
MaxPool1D Size	*2*
Conv2D activation	*ReLU*
Output Activation	*softmax*
Optimizer	*Adam*
Loss	*SCC*

	loss	accuracy	val_loss	val_accuracy
0	1.189192	0.546791	0.888876	0.685596
1	0.699524	0.767685	0.628770	0.802348
2	0.426931	0.878503	0.480725	0.842957
3	0.259817	0.939516	0.262416	0.938769
4	0.159511	0.971238	0.188955	0.962563
5	0.114062	0.980015	0.135424	0.976840
6	0.065054	0.992704	0.096758	0.985089
7	0.042953	0.996616	0.099908	0.982551
8	0.031719	0.997674	0.074131	0.986675
9	0.019681	0.998202	0.086057	0.986041
10	0.016538	0.998731	0.149400	0.969226
11	0.019724	0.998625	0.070287	0.988261
12	0.009439	0.999683	0.072023	0.988579
13	0.005027	1.000000	0.067441	0.989213
14	0.003959	1.000000	0.066436	0.987944

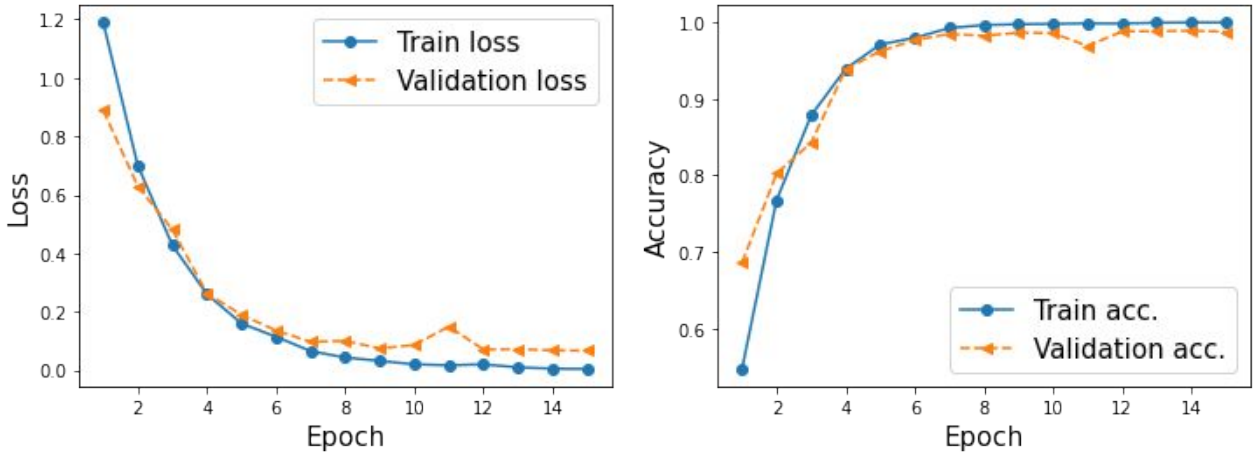
Berkeley
UNIVERSITY OF CALIFORNIA

Analyze Results: CNN_1D Classification Report

The results here are only slightly poorer than in CNN_2D. However, the grouping is still tight and with no class showing any obvious *bias* over another.

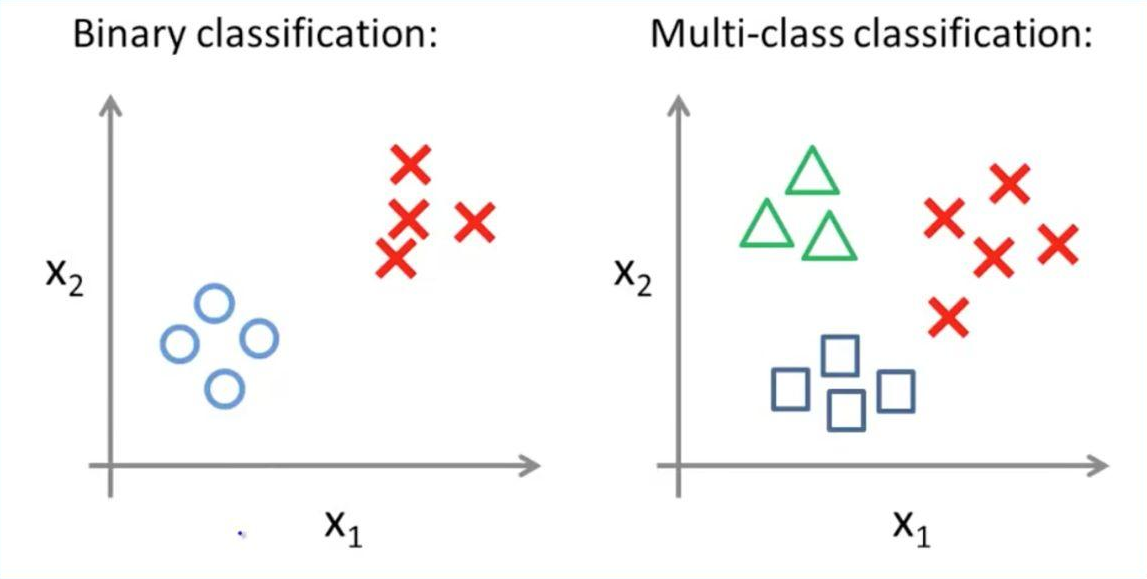
	precision	recall	f1-score	support
0	0.9890	0.9847	0.9869	458.0000
1	0.9941	0.9882	0.9911	508.0000
2	0.9814	0.9832	0.9823	537.0000
3	0.9906	0.9887	0.9897	533.0000
4	0.9786	0.9807	0.9796	466.0000
5	0.9847	0.9908	0.9877	651.0000
accuracy	0.9864	0.9864	0.9864	0.9864
macro avg	0.9864	0.9861	0.9862	3153.0000
weighted avg	0.9864	0.9864	0.9864	3153.0000

Analyze Results: CNN_1D Loss and Accuracy Plot



The plots here also show a converging pattern. But unlike in CNN_2D, the validation line show a bit of a bump at the 3rd and 11th epoch.

Multi-class Logistic Regression- Grayscale



Berkeley
UNIVERSITY OF CALIFORNIA

MCLR *Model Retrain* at optimum HPs over more epochs

HYPERPARAMETERS	VALUES
Epochs	**50** [10 - 60, step: 10]
Learning Rate	**0.001** [0.01, 0.001]
Batch Size	**32** [16, 32, 64]
Optimizer	**Adam** [SGD, Adam]
Output Activation	*softmax*
Loss	*Sparse Categorical Crossentropy (SCC)*

	loss	accuracy	val_loss	val_accuracy
0	2.321057	0.326319	2.214630	0.348985
1	1.672324	0.434387	1.581540	0.468274
2	1.367396	0.534842	1.484961	0.494607
3	1.180820	0.609073	0.907844	0.701777
4	1.017916	0.661521	1.179285	0.582487
45	0.072881	0.979380	0.146761	0.968274
46	0.227095	0.944697	0.113168	0.977792
47	0.064674	0.980226	0.132256	0.971129
48	0.355281	0.921857	1.055452	0.758883
49	0.083987	0.975256	0.108018	0.980013

Berkeley
UNIVERSITY OF CALIFORNIA

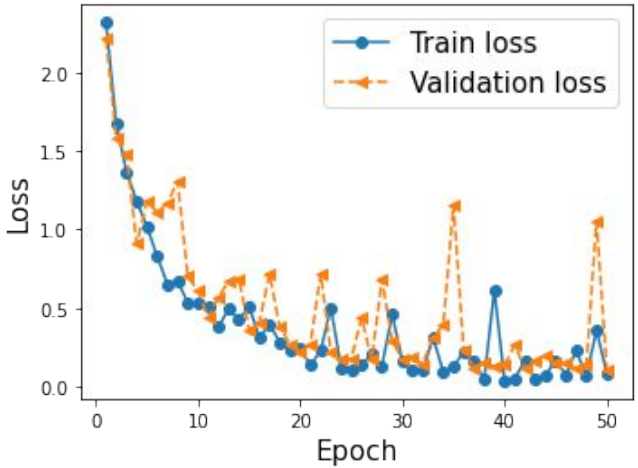
Analyze Results: MCLR Classification Report

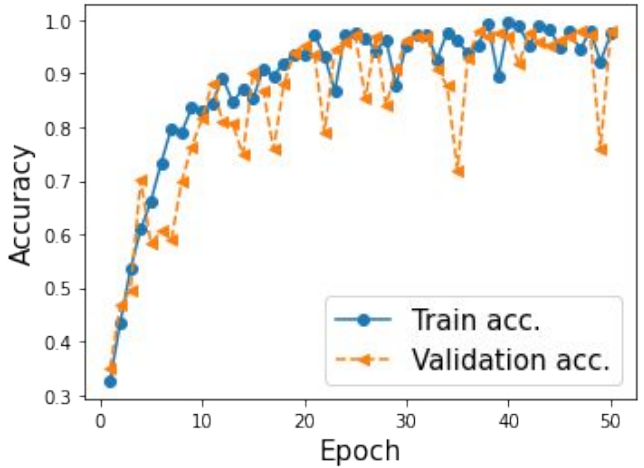
Although MCLR is the worst of the three in terms of loss and accuracy, the classification report still show no indication of bias and it still achieved a fairly decent accuracy score of 0.98.

	precision	recall	f1-score	support
0	0.9846	0.9782	0.9814	458.00
1	0.9980	0.9665	0.9820	508.00
2	0.9814	0.9832	0.9823	537.00
3	0.9637	0.9962	0.9797	533.00
4	0.9656	0.9635	0.9646	466.00
5	0.9862	0.9877	0.9870	651.00
accuracy	0.9800	0.9800	0.9800	0.98
macro avg	0.9799	0.9792	0.9795	3153.00
weighted avg	0.9802	0.9800	0.9800	3153.00

Berkeley
UNIVERSITY OF CALIFORNIA

Analyze Results: Multi-class Logistic Regression (MCLR) Loss and Accuracy Plot

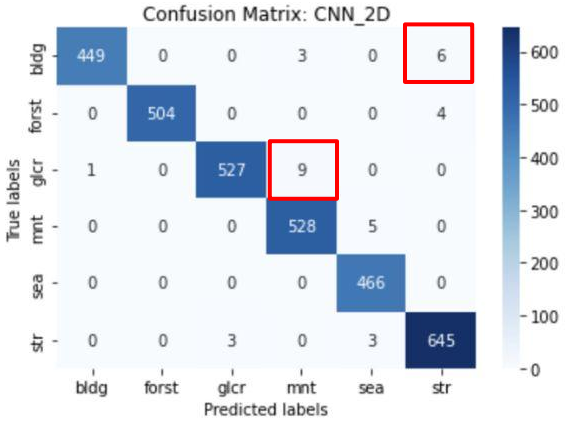


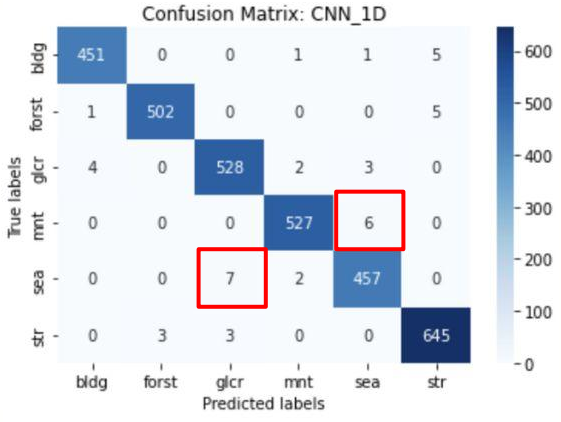


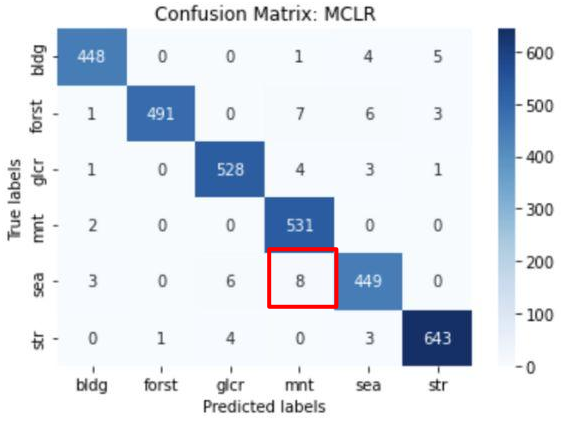
The path taken to achieve a > 0.98 score is erratic as shown above.

Berkeley
UNIVERSITY OF CALIFORNIA

Comparing Confusion Matrices







The confusion matrices vary in results. CNN_2D errors were mostly mountains vs. glaciers and street vs. buildings; CNN_1D: sea/glaciers and sea/mountain; MLCR: sea/mountain and forest/mountain

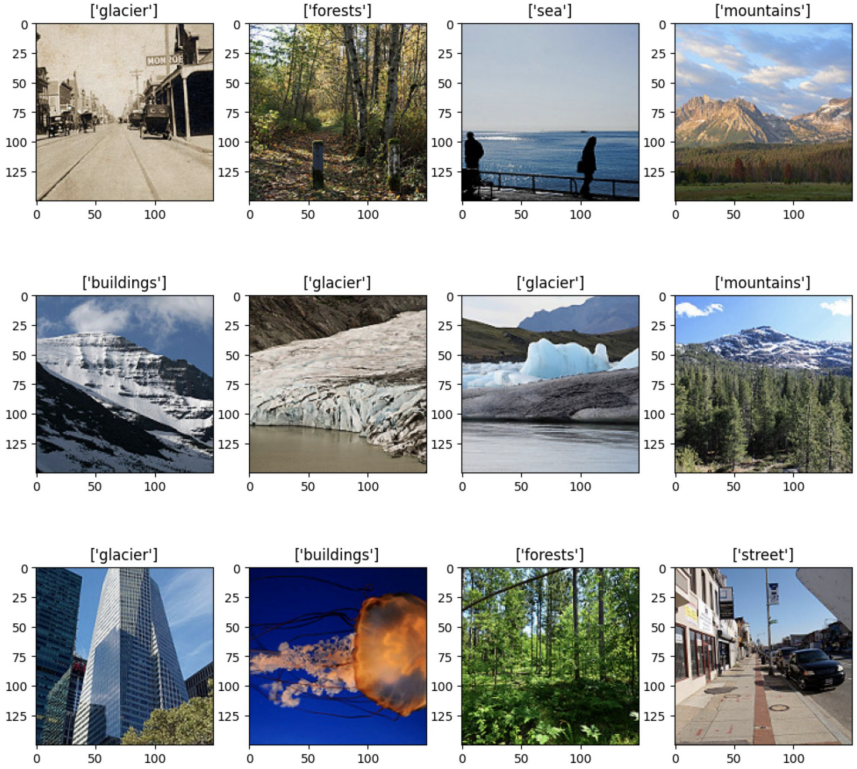
Conclusion: Model vs. Test Data

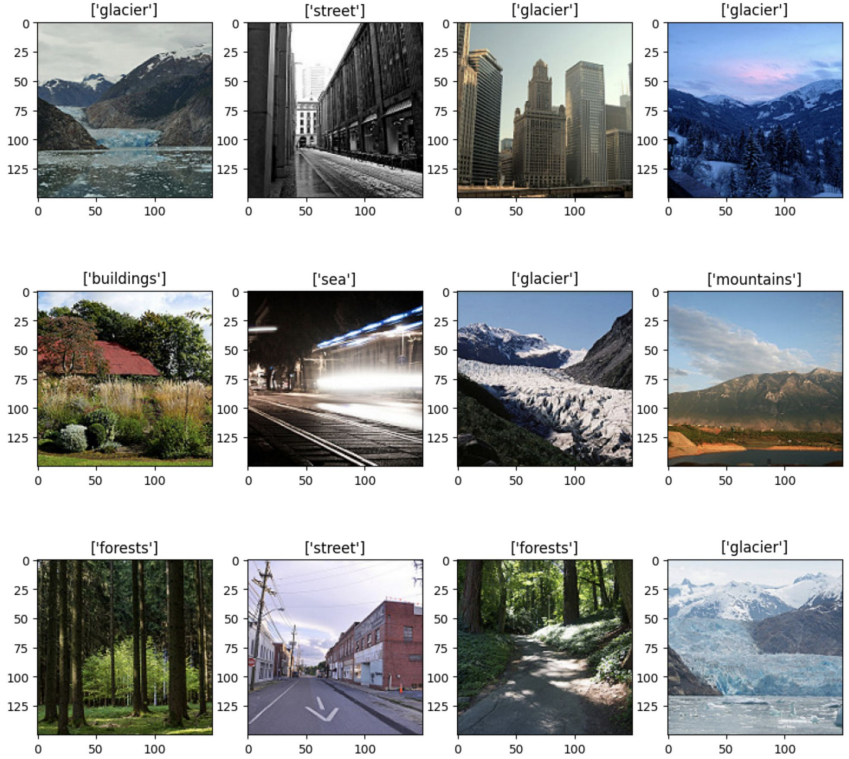
For our use case, model training will occur offline so training times don't matter. Therefore, due to its slightly higher classification accuracy, we would recommend a 2D CNN for classifying images in a production environment.

model.evaluate( )	Loss	Accuracy
CNN_2D	**0.030794**	**0.992705**
CNN_1D	**0.066340**	**0.986362**
MCLR	**0.150250**	**0.980019**

Berkeley
UNIVERSITY OF CALIFORNIA

Conclusion: Final Test





Thank you.

Berkeley
UNIVERSITY OF CALIFORNIA