

# DSSC - CVPR

Grimaldi Marco Alberto

February 2020

## Introduction

This project requires the implementation of an image classifier based on convolutional neural networks. It's divided in three parts which I'll cover one at time. The data-set is given and has been already split into training and test sets. I used MATLAB.

### Common Setup

For all steps and scripts there are some common features like:

- Input resizing: to match different CNN 1<sup>st</sup> layer proprieties (in all three dimensions), I used "imresize", "repmat" and "augmentedImageDatastore" option.
- Input normalization: is almost always automatically done by 1<sup>st</sup> layer.
- Data augmentation: to avoid overfitting and improve performance; I used translations, rotations, shear and reflection along vertical axe.
- Early stopping criteria: also to avoid overfitting, I used "MaxEpoch" and/or "Patience" on Validation Set Loss.
- Train-Validation split: 85%-15%.

## Part 1

Just followed given instruction, only interesting part is recognize that (without using data augmentation) after a couple of epoch Validation Loss start to increase while Training Loss is still decreasing (same but reversed on accuracy score), meaning that the CNN is no more learning features but just overfitting on the Training Set.

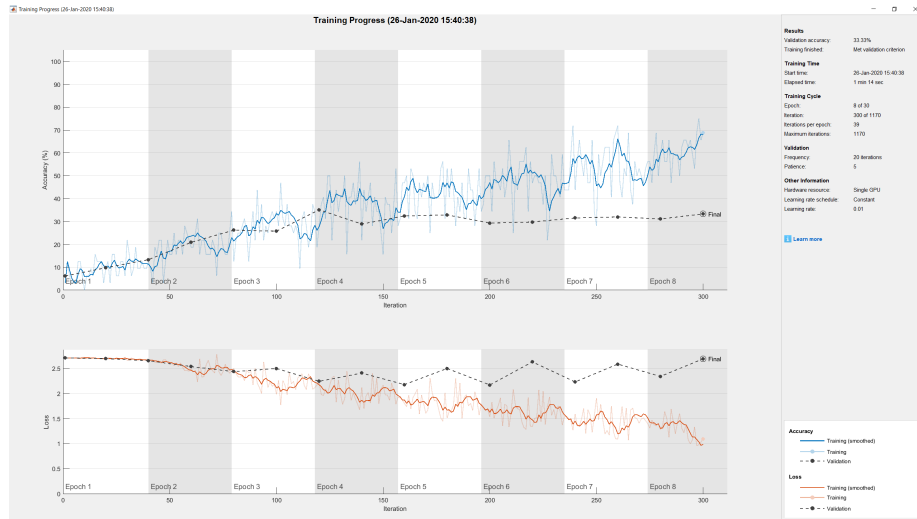


Figure 1: Shallow NN

		Confusion Matrix															
Output Class		Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding	
		21	8	6	3	9	1	5	8	24	7	6	0	1	4	16	17.6%
Bedroom		0.7%	0.3%	0.2%	0.1%	0.3%	0.0%	0.2%	0.3%	0.8%	0.2%	0.2%	0.0%	0.0%	0.1%	0.5%	82.4%
		2	109	6	7	0	0	0	0	10	2	55	0	0	0	2	56.5%
Coast		0.1%	3.7%	0.2%	0.2%	0.0%	0.0%	0.0%	0.0%	0.3%	0.1%	1.8%	0.0%	0.0%	0.0%	0.1%	43.5%
		0	2	28	3	2	7	3	0	5	4	11	3	1	1	3	38.4%
Forest		0.0%	0.1%	0.9%	0.1%	0.1%	0.2%	0.1%	0.0%	0.2%	0.1%	0.4%	0.1%	0.0%	0.0%	0.1%	61.6%
		2	49	0	116	6	1	0	1	8	1	32	0	1	0	2	63.0%
Highway		0.1%	1.6%	0.0%	3.9%	0.2%	0.0%	0.0%	0.0%	0.3%	0.0%	1.1%	0.0%	0.0%	0.0%	0.1%	47.0%
		12	3	12	4	78	25	7	21	28	18	12	13	9	13	28	27.6%
Industrial		0.4%	0.1%	0.4%	0.1%	2.6%	0.8%	0.2%	0.7%	0.9%	0.6%	0.4%	0.4%	0.3%	0.4%	0.9%	72.4%
		5	2	18	0	4	45	1	1	5	1	5	18	9	4	6	36.3%
InsideCity		0.2%	0.1%	0.6%	0.0%	0.1%	1.5%	0.0%	0.0%	0.2%	0.0%	0.2%	0.6%	0.3%	0.1%	0.2%	63.7%
		23	10	21	4	15	21	25	35	35	7	11	23	8	10	19	9.4%
Kitchen		0.8%	0.3%	0.7%	0.1%	0.5%	0.7%	0.8%	1.2%	1.2%	0.2%	0.4%	0.8%	0.3%	0.3%	0.6%	90.6%
		38	13	40	5	35	38	36	93	66	28	33	63	38	49	57	14.7%
LivingRoom		1.3%	0.4%	1.3%	0.2%	1.2%	1.3%	1.2%	3.1%	2.2%	0.9%	1.1%	2.1%	1.3%	1.6%	1.9%	85.3%
		3	11	3	3	3	1	4	1	21	1	15	0	0	2	1	30.4%
Mountain		0.1%	0.4%	0.1%	0.1%	0.0%	0.1%	0.0%	0.7%	0.0%	0.5%	0.0%	0.0%	0.0%	0.1%	0.0%	69.6%
		1	10	50	4	12	11	9	6	14	38	11	4	3	0	30	18.7%
Office		0.0%	0.3%	1.7%	0.1%	0.4%	0.4%	0.3%	0.2%	0.5%	1.3%	0.4%	0.1%	0.1%	0.0%	1.0%	81.3%
		0	40	3	8	6	2	2	0	24	0	103	0	2	0	1	53.9%
OpenCountry		0.0%	1.3%	0.1%	0.3%	0.2%	0.1%	0.1%	0.0%	0.8%	0.0%	3.5%	0.0%	0.1%	0.0%	0.0%	46.1%
		6	0	26	1	12	29	8	11	8	0	4	72	19	15	9	32.7%
Store		0.2%	0.0%	0.9%	0.0%	0.4%	1.0%	0.3%	0.4%	0.3%	0.0%	0.1%	2.4%	0.6%	0.5%	0.3%	67.3%
		1	0	7	0	5	15	4	1	5	0	5	6	95	8	6	60.1%
Street		0.0%	0.0%	0.2%	0.0%	0.2%	0.5%	0.1%	0.0%	0.2%	0.0%	0.2%	3.2%	0.3%	0.2%		39.9%
		0	1	1	1	1	3	2	2	3	0	1	1	6	35	1	60.3%
Suburb		0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.1%	0.1%	0.0%	0.0%	0.0%	0.2%	1.2%	0.0%		39.7%
		2	2	7	1	23	9	4	9	18	8	6	12	0	0	75	42.6%
TallBuilding		0.1%	0.1%	0.2%	0.0%	0.8%	0.3%	0.1%	0.3%	0.6%	0.3%	0.2%	0.4%	0.0%	0.0%	2.5%	57.4%
		18.1%	41.9%	12.3%	72.5%	37.0%	21.6%	22.7%	49.2%	7.7%	33.0%	33.2%	33.5%	49.5%	24.8%	29.3%	32.0%
		81.9%	58.1%	87.7%	27.5%	63.0%	78.4%	77.3%	50.8%	92.3%	67.0%	66.8%	66.5%	50.5%	75.2%	70.7%	68.0%
		Target Class															
		Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding	

Figure 2: Shallow NN

The overall validation accuracy is around 33%. Test accuracy around 32%.

## Part 2

For that part the goal was to improve the CNN with some different techniques. In addition to data-augmentation I added a couple more of convolutional layers and also introduced batch-normalization and dropout layers.

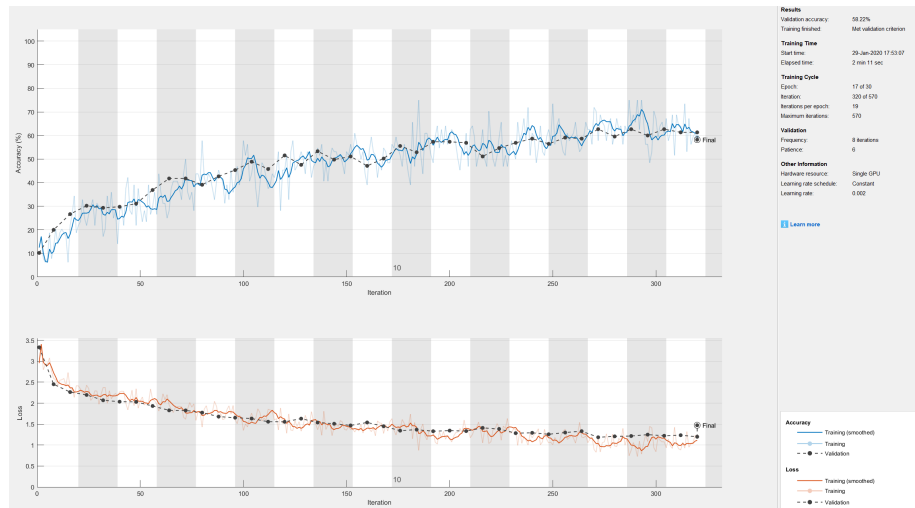


Figure 3: Improved NN

Bedroom	46			2	7	2	8	29	13	1	1	2			5
Coast		133	3	51		1			13		58		1		
Forest			204		4				5		7	7	1		
Highway	2	2	5	130	3				7		5	3	3		
Industrial	8		11	10	103	9	5	6	4	1	14	24	8	1	7
InsideCity	1	1	8	1	16	107	3	5		2	3	36	21	1	3
Kitchen	12			4	8	4	38	19	2	4	1	10	4		4
LivingRoom	26		1	2	12	3	20	102	3	2	1	10	5	1	1
Mountain		2	20	5	2				200		41	2	2		
Office	19		4	1	16	8	5	12	8	37		2			3
OpenCountry		17	17	22	3				41		207		3		
Stone			15		9	8	3	5	3	1		161	6	1	3
Street			7	5	6	1	1				6	6	159		1
Suburb	1		6	2	5	7	1	3			4	5	28	79	
TallBuilding	1		11		49	7			2	4	1	2	18	2	159
	Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Stone	Street	Suburb	TallBuilding

Figure 4: Improved NN

The overall validation accuracy is around 58% and test accuracy is around 57%. Notice how all the new features prevent the CNN to overfit the training set, reaching a kind of plateau with train and validation losses lines really close to each other.

I've also tried to use an ensemble of 10 of the previous NN.

True Class	Bedroom	67	2	2	2	4	1	13	19	5	2		1	1	1	2
	Coast		171	2	25		1			2		59				
	Forest			358		2				19			2	2		
	Highway		12	4	122	1	2			4		10	2	3		
	Industrial	7	4	19	8	85	12	5	2	15	4	13	19	6	2	12
	InsideCity			16	4	5	106	5	1	4	4	8	26	20	3	2
	Kitchen	11		1	4	5	4	54	5	2	5	1	9	6		3
	LivingRoom	29	1	2		17	4	45	48	6	4	3	13	8	1	8
	Mountain		9	17	3				214			26		2		1
	Office	10	1	5		8	9	7	7	3	51		2			12
	OpenCountry		19	16	16					34		223		1	1	
	Store			44		4	7	2		4			144	5	2	3
	Street			14	3	2	5			2		5	7	103		2
	Suburb			9	1	2	7		2	4		4	6	15	91	
	TallBuilding	1		12		23	5	1	3	8	1	1	10	2		150
		Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom Predicted Class	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding

Figure 5: Ensemble of improved NN

I've reached a test accuracy a bit over 64% using the arithmetic means of the output classification score from each CNN.

## Part 3

Lastly was required to adopt a Transfer Learning approach importing using trained CNNs. There are two approach for that:

- Replace last Full-Connected layer with another one (with the right dimension for our classification problem) and then fine-tune the derived CNN.
- Get activators from last Convolutional layer and then train a SVM to get classification.

I tried first approach with AlexNet and VGG, for AlexNet I tried both freezing all layers except the new one and no freeze but with 0.1x of learning rate and 20x LearnFactor on last layer. For VGG the only possible option due HW limitation was the first one.

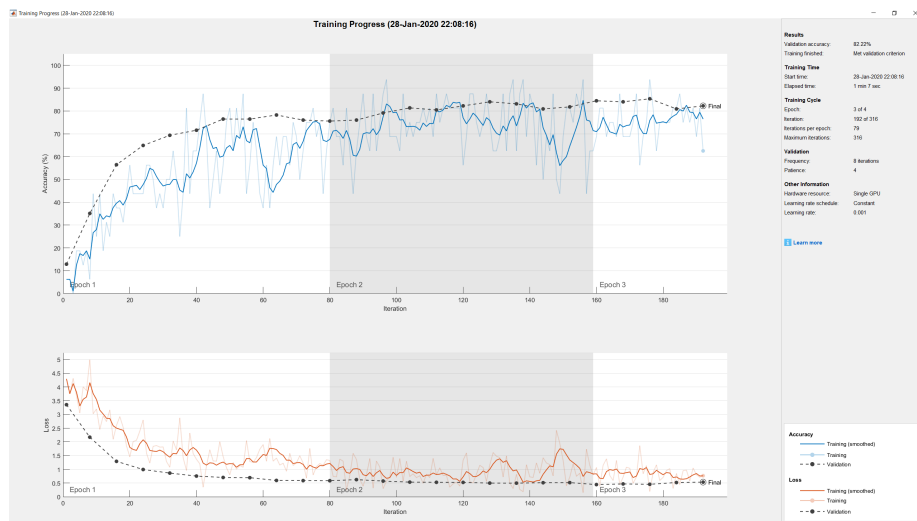


Figure 6: AlexNet Freeze

Bedroom	94					2	9	16		5							
Coast		100	1	9					1		50					1	
Forest			300						2		19				2		
Highway		1		139	4	4		1			4	2	1	3	1		
Industrial				1	144	43	2	1	1	1			1	9	9		
InsideCity						159	2							1	6		
Kitchen	2					3	100			4			1				
LivingRoom	17				2	12	56	89		13						2	
Mountain		1	3	1						140		28				1	
Office						2	19	5			80						
OpenCountry			16	1	5	2	1			10		279				3	
Store	2		1			15	37	12	2	4	2		136	1			3
Street				4	4	46	2	1					1	127	3	1	
Suburb						4		1							138		
TallBuilding			1			20	30			1						204	
	Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding		

Figure 7: AlexNet Freeze

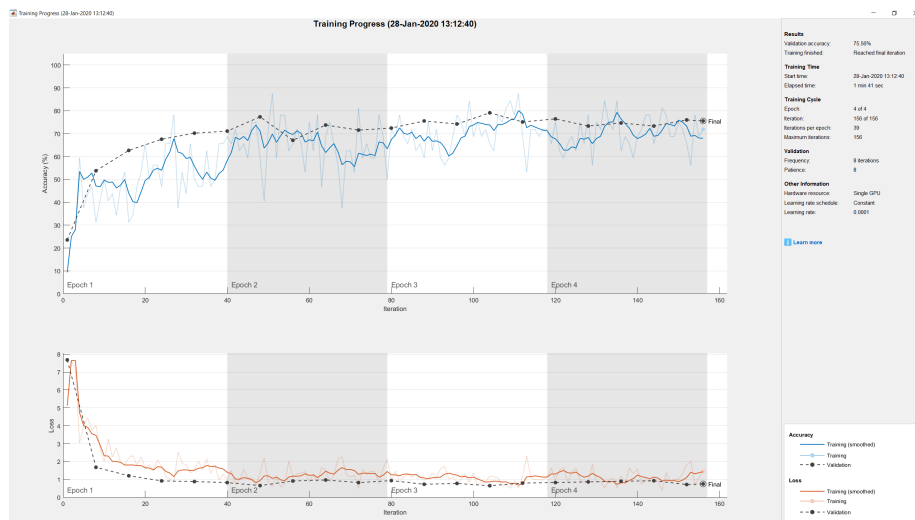


Figure 8: AlexNet No Freeze

Bedroom	76				1		2	36				1			1
Coast	1	223	1	2	2	1			1		29				
Forest		1	214						7		6				
Highway		4		133	7	4	1	1			5	1	3		1
Industrial	2			4	140	13	5	8	10		3	10	1		15
InsideCity	3				19	176	7	2				1	2	2	2
Kitchen	7					1	90	10			1				
LivingRoom	31			1	3		14	130	1		1	3	1		4
Mountain		7	5	2					241		14				
Office	15						17	27		52		4			
OpenCountry		40	7	4	3				17		279				
Store	4		1	5	7	13	22	10	2			145	4		2
Street				9	8	22	3		1				142		7
Suburb	2		4		4	4		2	1		2			120	2
TallBuilding			1		13	3			3			1	1		254
	Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding

Figure 9: AlexNet No Freeze

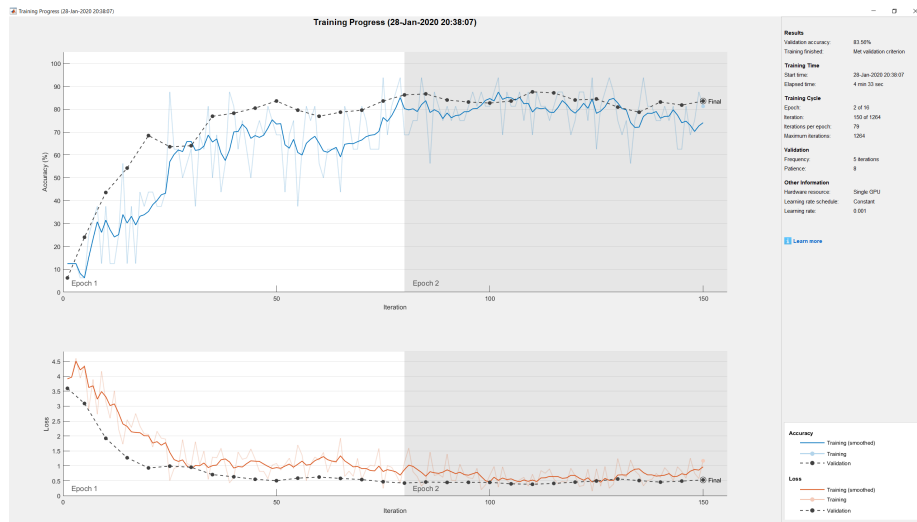


Figure 10: VGG

For second approach I used Resnet18 and Error-Correcting Output Codes Model for training the SVM classifier.

	Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding
Bedroom	95						2	19							
Coast		227	1	3	1								17		
Forest			214						1				13		
Highway				147	5	1						1		6	
Industrial				2	102	9						1	2	1	3
InsideCity					11	162	1			1				10	2
Kitchen	1					1	102	3		1		2			
LivingRoom		22					1	16	145		2			1	
Mountain			2	5	2				254		11				
Office							2	3		110					
OpenCountry									24		249				1
Store					6	6	4	3		1		195			
Street				2	4	6						1	177		2
Suburb				1	1	5								134	
TallBuilding			1		12	9							3		211
	Bedroom	Coast	Forest	Highway	Industrial	InsideCity	Kitchen	LivingRoom	Mountain	Office	OpenCountry	Store	Street	Suburb	TallBuilding

Figure 11: SMV

All approaches went beyond 75% accuracy, with the SVM almost crossing 90% accuracy test score. Notice also that no overfitting is present (which can be a common problem when dealing with transfer learning due to the imbalance between the huge prior training set and the new one) this is mostly thanks to data augmentation applied on training set and early-stopping criterion on training routine.