DATA 442: Neural Networks & Deep Learning

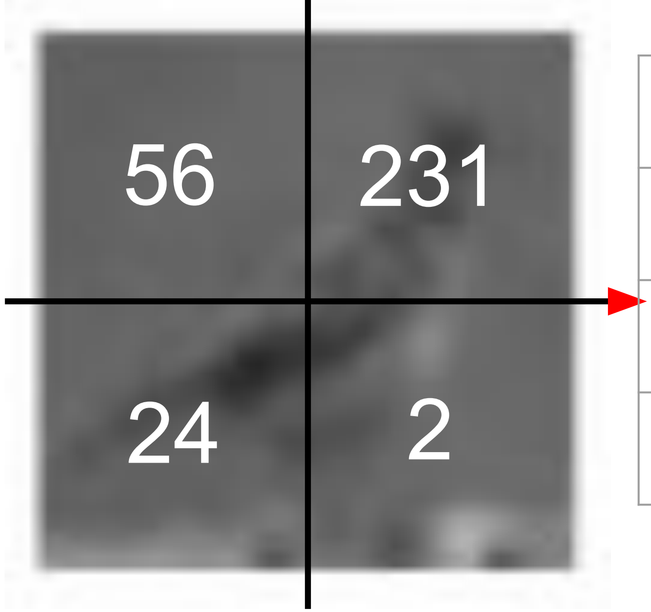
Dan Runfola – danr@wm.edu

icss.wm.edu/data442/

```
def predict(image, W):
    W*image
```

nn.predict(image, **W**)

	Probability
Bird	0.2
Dog	0.1
...	...
Cat	0.15
Plane	0.19



56
231
24
2

0.2	-0.5	0.1	2.0	Cat
1.5	1.3	2.1	0.0	Bird
0	0.25	0.2	-0.3	Plane

Cat Score = (56 * 0.2) + (231 * -0.5) + (24 * 0.1) + (2 * 2.0) = -97.9

```
def predict(image, W):
    W*image
```

Cat Score = -97.9

Bird Score = 434.7

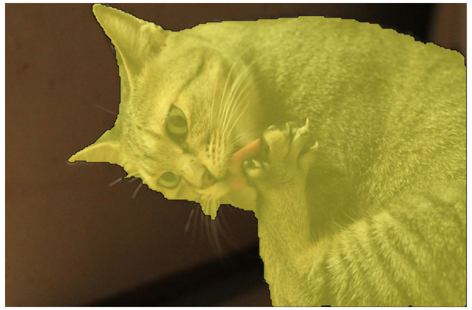
Plane Score = 63.15

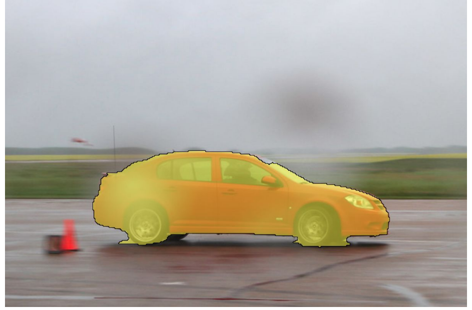
$$\text{Total Loss} = \frac{1}{N}\sum_i^N Loss_i(f(x_i, W), y_i)$$

where ***N*** is the total number of images (i.e., 3), ***i*** is a unique index for each image, ***x_i*** is the image itself, ***y_i*** is the image label, ***Loss_i*** is the loss for that image, and **W** is the weights being tested.

f(image, W) = scores

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1





Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

Multiclass SVM Loss

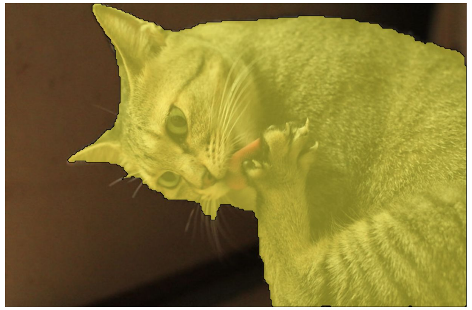
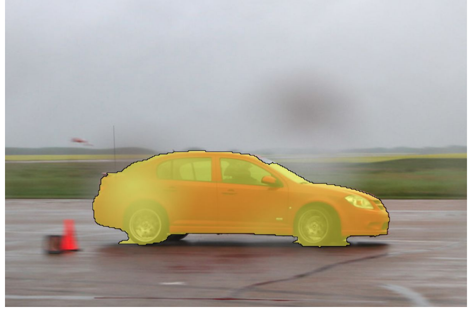
J is the total number of classes, represented by index *j*. In the current example, *j=1* would be “Cat”, *j=2* would be “Car”, etc.

s is the score for a given category. For the first image (the Cat), s_1 would be 3.2, s_2 would be 5.1, and s_3 would be -1.7.

Epsilon (**ε**) is a tolerance term, essentially defining how sure the algorithm needs to be about a class before we call it right.

$$\sum_{j \neq y_i}^{J} max(0, s_j - s_{y_i} + \varepsilon)$$

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1

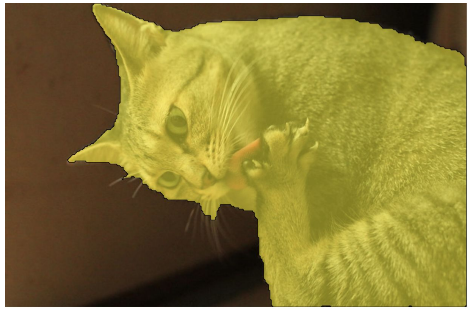

Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

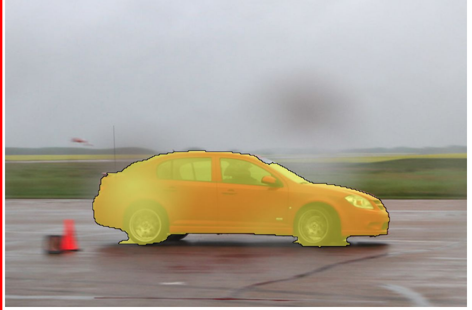
Multinomial Logistic Regression - Softmax

<u>Probability the picture is a</u>
Cat: 34%
Car: 51%
Frog: 15%

Softmax Function

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1





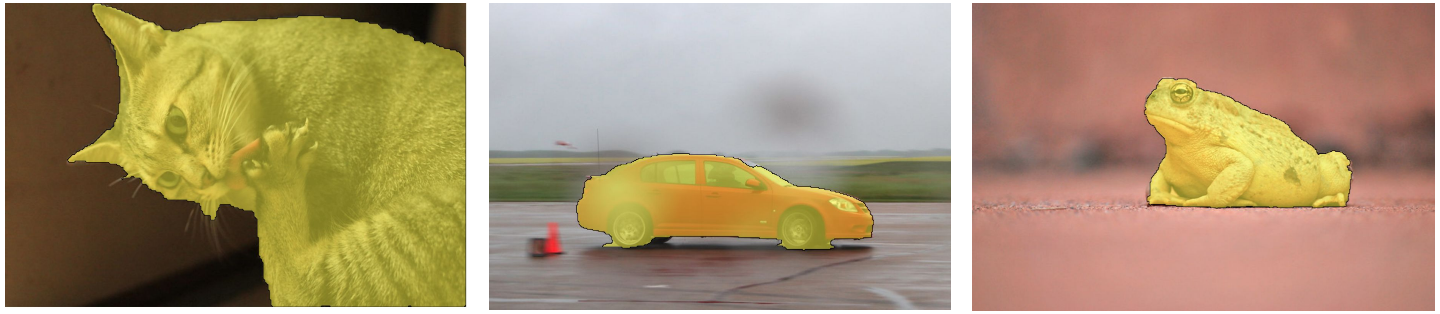
Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

Multinomial Logistic Regression - Softmax

Assumption: These are really probabilities, just unnormalized!

Specific Assumption: These are unnormalized log probabilities for each class.

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1



Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

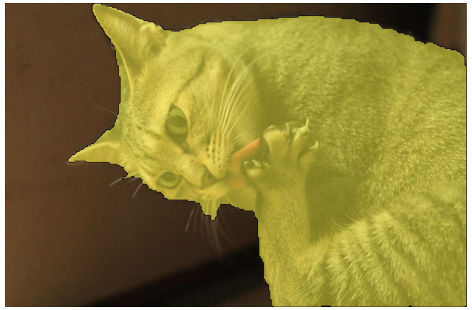
Multinomial Logistic Regression - Softmax

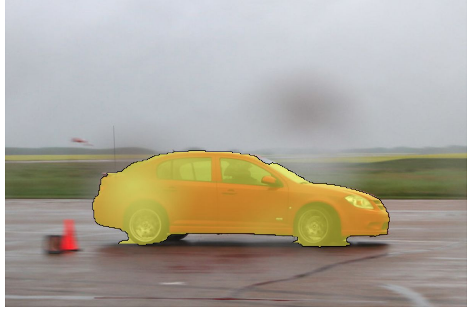
$$P(Y = k | X = X_i)$$

Assumption: These are really probabilities, just unnormalized!

Specific Assumption: These are unnormalized log probabilities for each class.

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1





Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

Multinomial Logistic Regression - Softmax

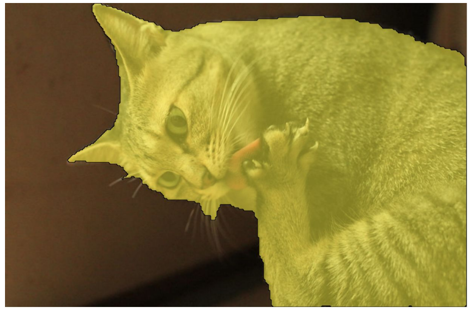
$$P(Y = k | X = X_i) = \frac{e^{s_k}}{\sum_{j=1}^{J} e^{s_j}}$$

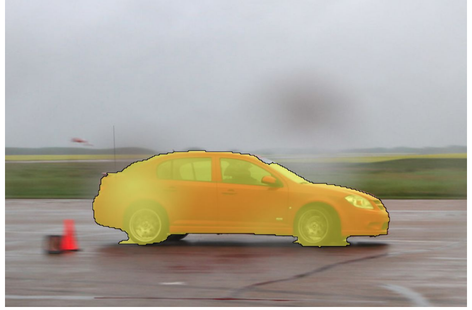
This is the softmax function for class k.

Assumption: These are really probabilities, just unnormalized!

Specific Assumption: These are unnormalized log probabilities for each class.

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1





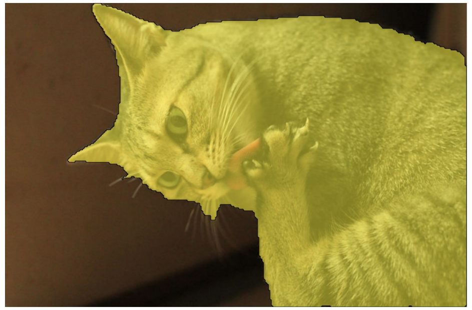
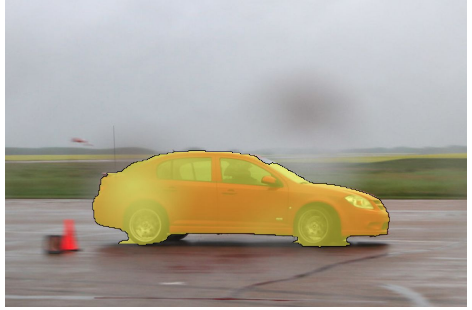
Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

Multinomial Logistic Regression - Softmax

$$\frac{e^{s}_{k}}{\sum_{j=1}^{J} e^{s}_{j}}$$

In a perfect world for this example, the above function would result in 1 for cat, and 0 for both car and frog.

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1

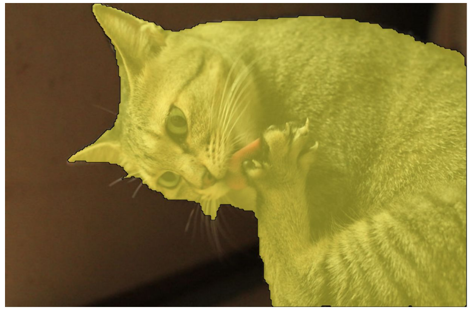



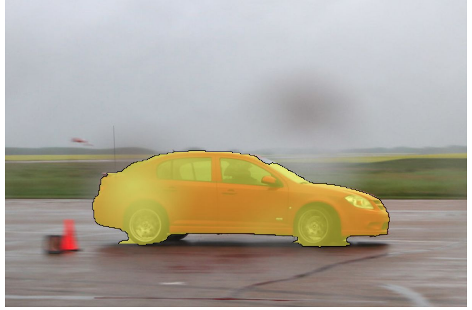

Multinomial Logistic Regression - Softmax

$$Loss_i = -1 * \frac{e^s_k}{\sum_{j=1}^J e^s_j}$$

$$\frac{e^s_k}{\sum_{j=1}^J e^s_j}$$

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1



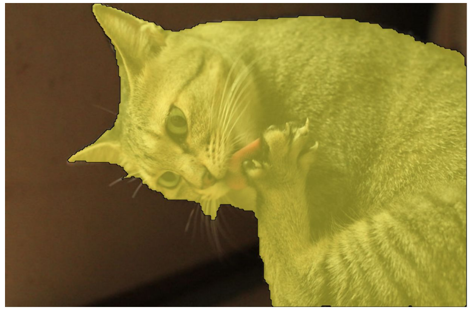
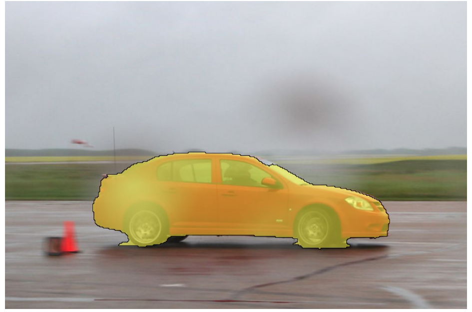


Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

$$Loss_i = -1 * log(\frac{e^s_k}{\sum_{j=1}^{J} e^s_j})$$

$$\frac{e^s_k}{\sum_{j=1}^{J} e^s_j}$$

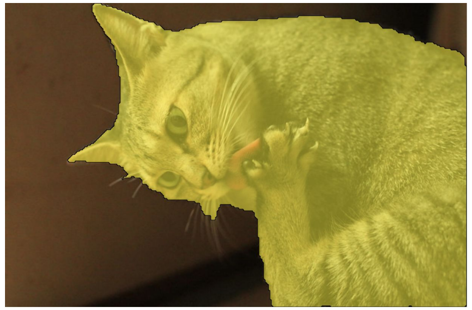
Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1

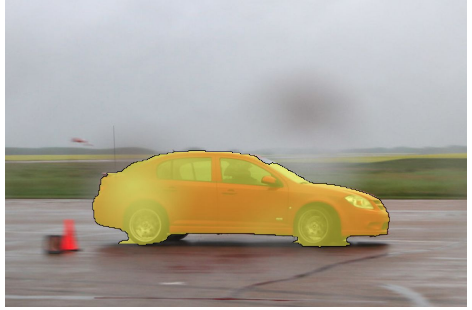




Multinomial Logistic Regression - Softmax

$$L_i = -log(\frac{e^s_k}{\sum_{j=1}^{J} e^s_j})$$

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1





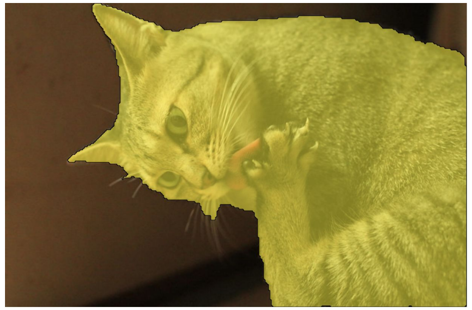
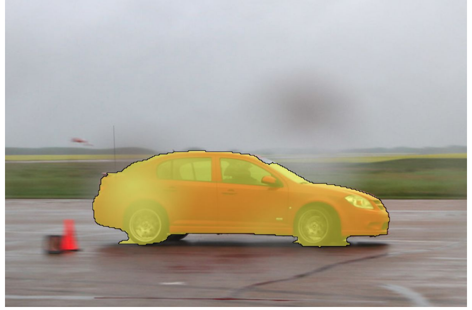
Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

Multinomial Logistic Regression - Softmax

Class	Score	e^s
Cat	3.2	24.5
Car	5.1	164.0
Frog	-1.7	0.18

$$L_i = -log(\frac{\boxed{e^{s}_{k}}}{\sum_{j=1}^{J} e^{s}_{j}})$$

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1


Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

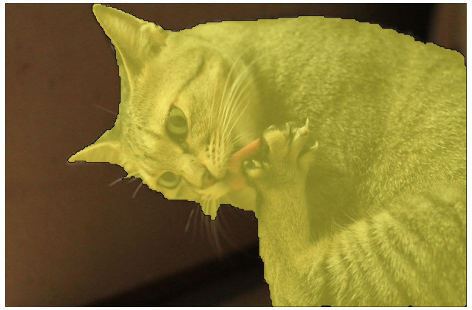
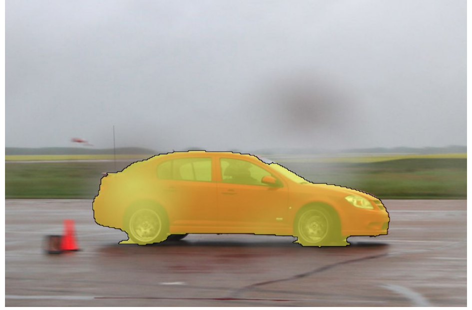
Multinomial Logistic Regression - Softmax

Class	Score	e^s	e^s/188.68
Cat	3.2	24.5	0.13
Car	5.1	164.0	0.87
Frog	-1.7	0.18	0.00

188.68

$$L_i = -log(\frac{e^s_k}{\boxed{\sum_{j=1}^{J} e^s_j}})$$

Cat	**3.2**	**1.3**	**2.2**
Car	**5.1**	**4.9**	**2.5**
Frog	**-1.7**	**2.0**	**-3.1**


Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

Multinomial Logistic Regression - Softmax

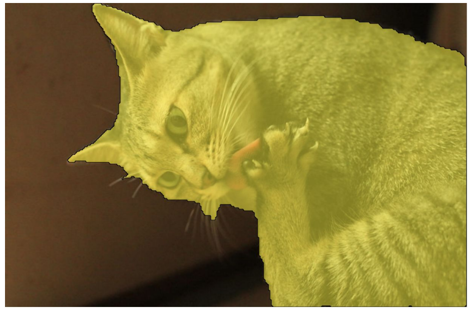
$$-log(0.13) = 0.89$$

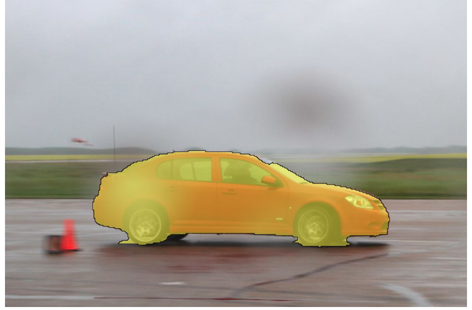
Class	Score	e^s	e^s/188.68
Cat	3.2	24.5	0.13
Car	5.1	164.0	0.87
Frog	-1.7	0.18	0.00

188.68

$$L_i = \boxed{-log(\frac{e^s_k}{\sum_{j=1}^{J} e^s_j})}$$

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1



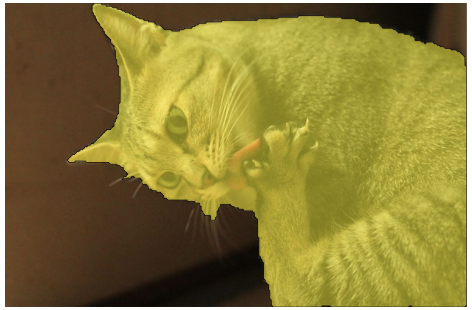


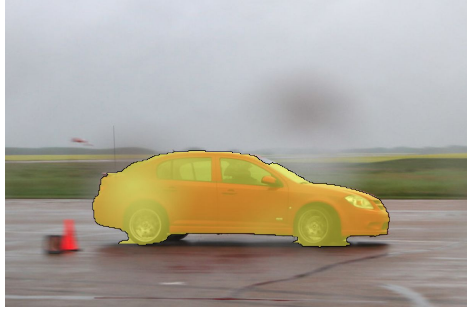
Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

Multinomial Logistic Regression - Softmax

$$L_i = -log(\frac{e^s_k}{\sum_{j=1}^{J} e^s_j})$$

Loss_i	0.89	0.034	2.67
Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1





Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park

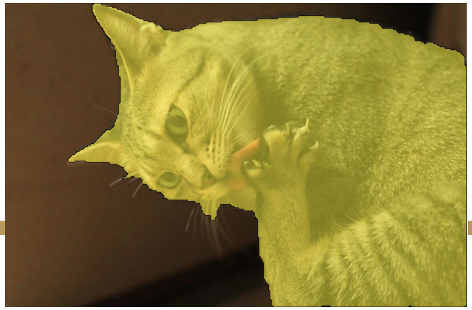
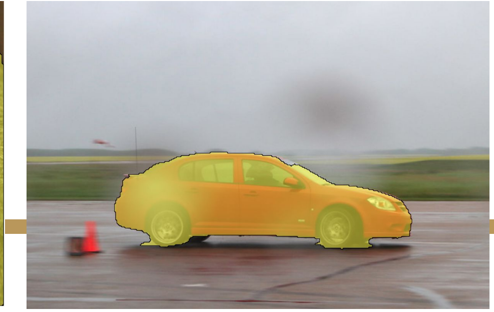
Multiclass SVM Loss

$$\sum_{j \neq y_i}^{J} max(0, s_j - s_{y_i} + \varepsilon)$$

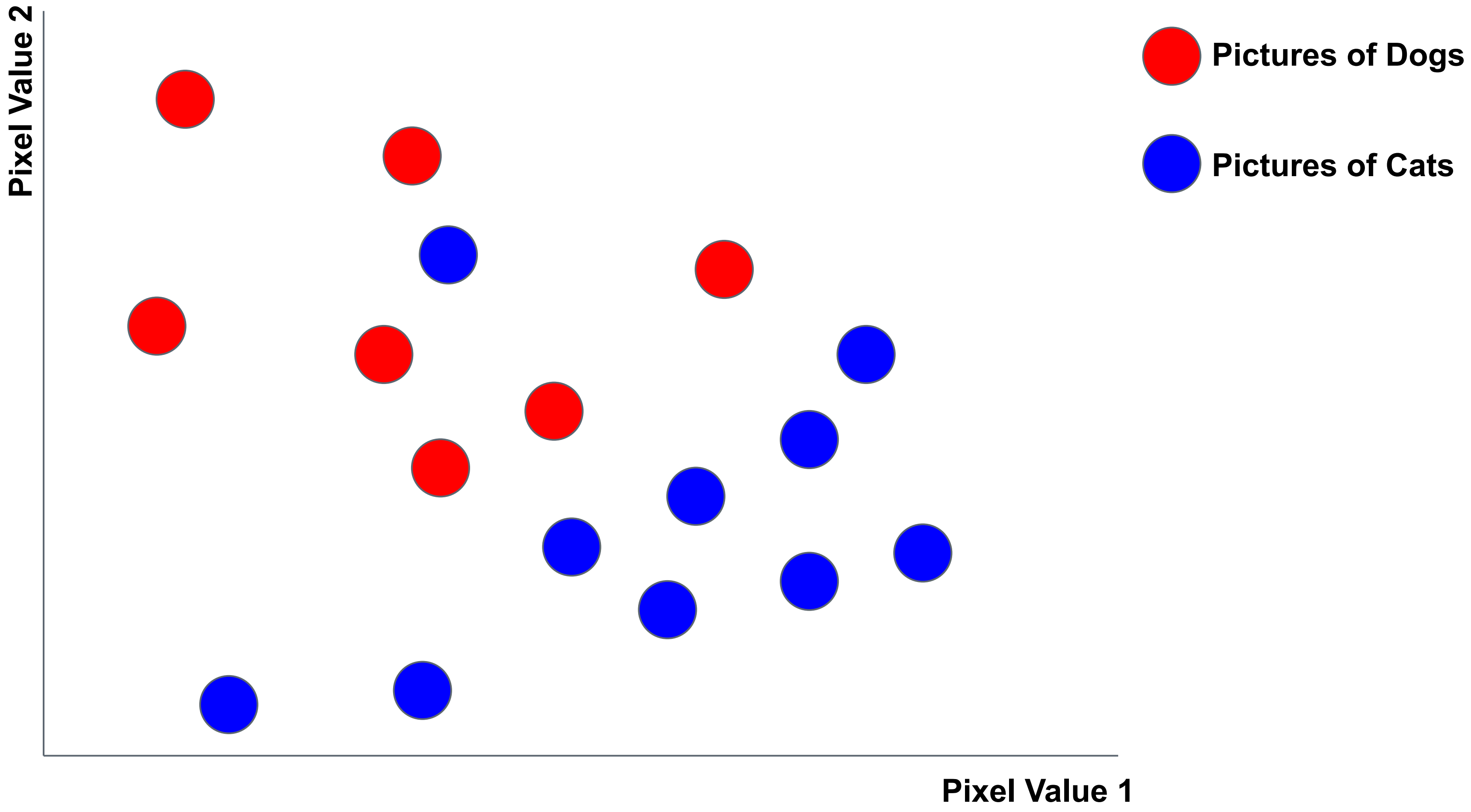
Multinomial Logistic Regression - Softmax

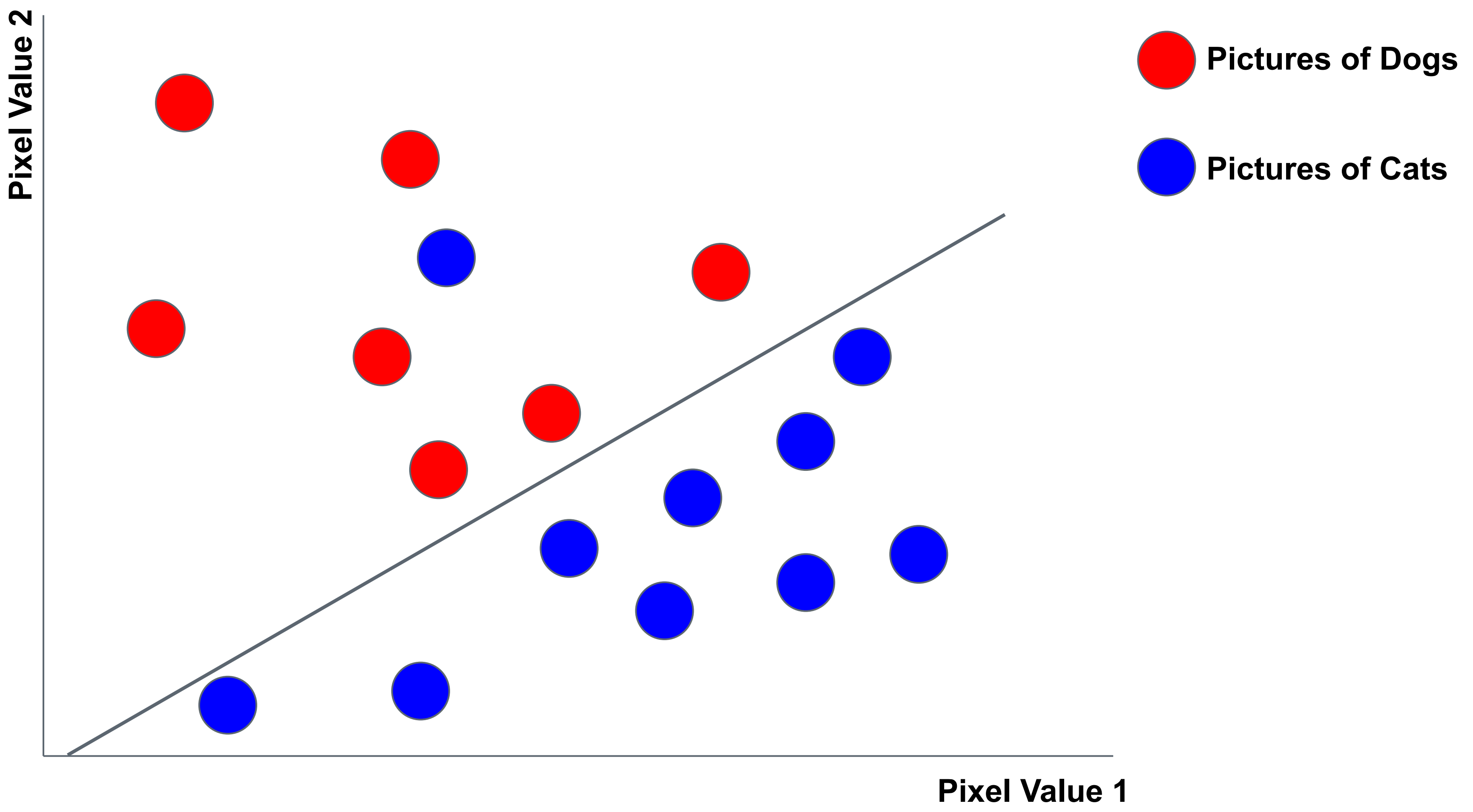
$$L_i = -log(\frac{e^s_k}{\sum_{j=1}^{J} e^s_j})$$

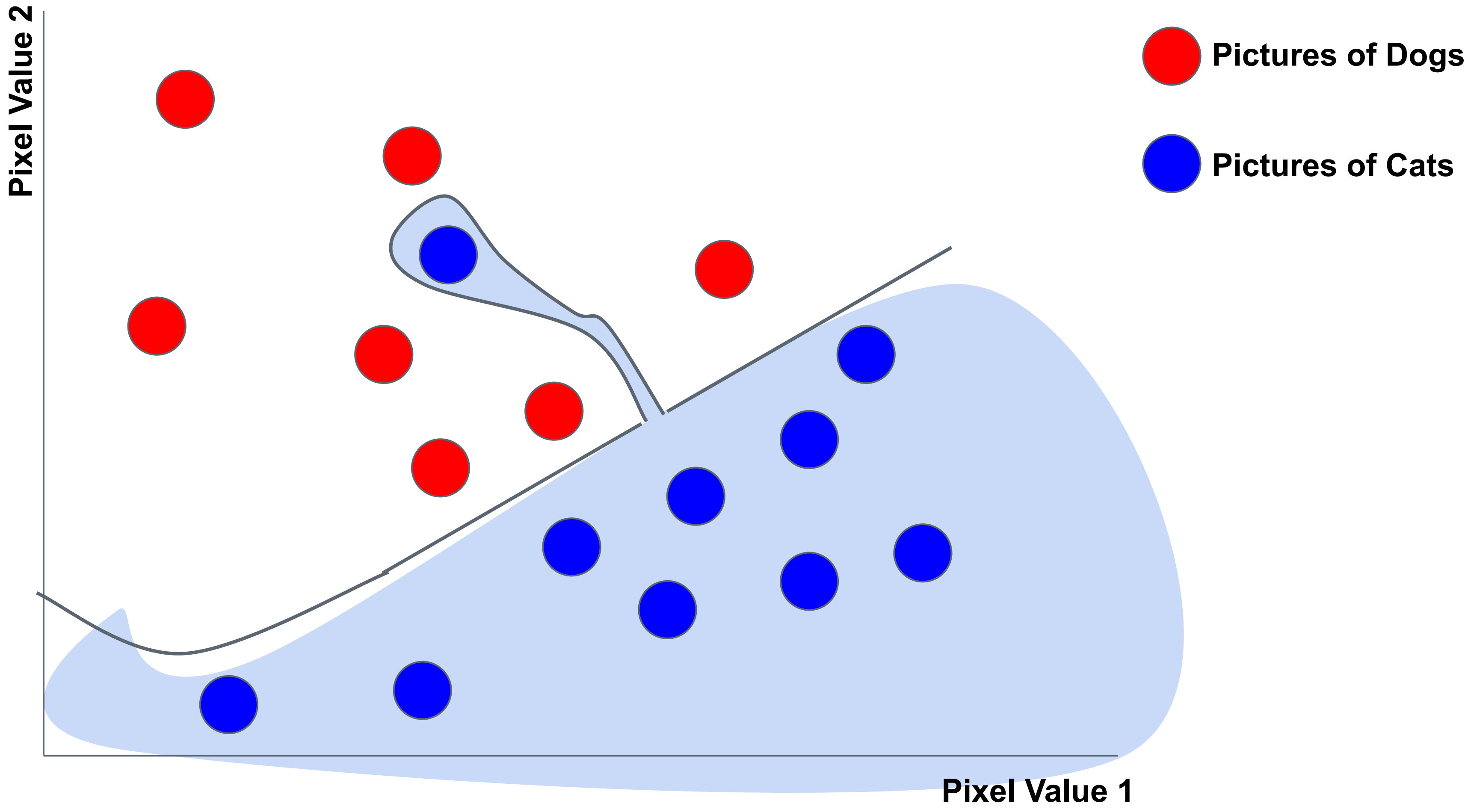
SVM	**2.9**	**0**	**12.9**
Softmax	**0.89**	**0.034**	**2.67**
Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1

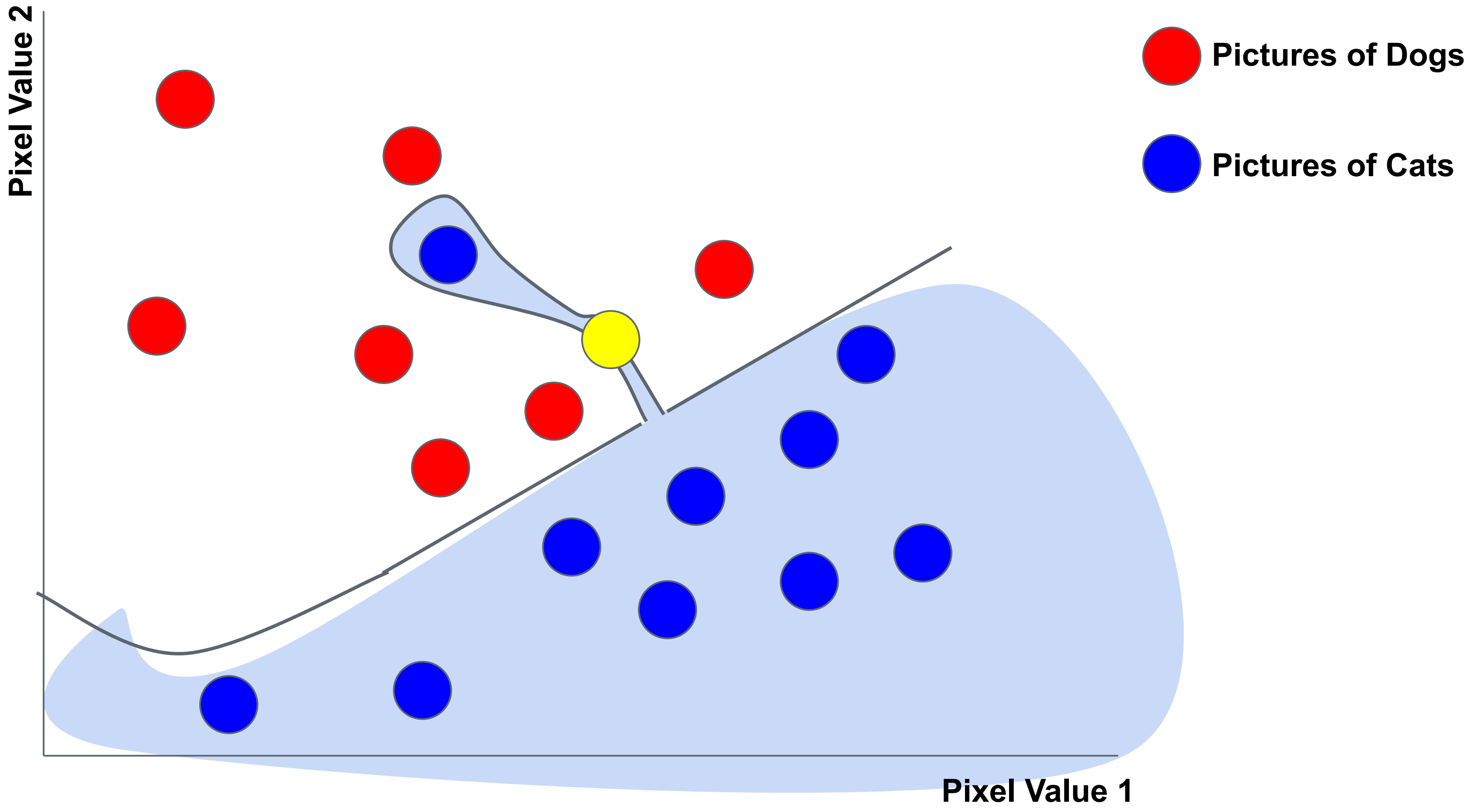




Images CC Attribution, Left to Right: Yutaka Fujiki, Grant C., Zion National Park


Pixel Value 2
Pixel Value 1
Pictures of Dogs
Pictures of Cats


Pixel Value 2
Pixel Value 1
Pictures of Dogs
Pictures of Cats


Pixel Value 2
Pixel Value 1
Pictures of Dogs
Pictures of Cats


Pixel Value 2
Pictures of Dogs
Pictures of Cats
Pixel Value 1

$$\underbrace{\frac{1}{N}\sum_{i}^{N} Loss_i(f(x_i, W), y_i)}_{\text{Total Loss (Data Loss)}}$$

$$\underbrace{\frac{1}{N}\sum_i^N Loss_i(f(x_i, W), y_i)}_{\text{Data Loss}} + \underbrace{\lambda R(W)}_{\text{Regularization Loss}}$$

$$\underbrace{\frac{1}{N}\sum_{i}^{N} Loss_i(f(x_i, W), y_i)}_{\text{Data Loss}} + \underbrace{\boxed{\lambda} R(W)}_{\text{Regularization Loss}}$$

$$\frac{1}{N}\sum_i^N Loss_i(f(x_i, W), y_i) + \lambda \boxed{R(W)}$$

L2 Regularization $$R(W) = \sum_{k=1}^{K} W_k^2$$

$$\frac{1}{N}\sum_{i}^{N} Loss_i(f(x_i, W), y_i) + \lambda \boxed{R(W)}$$

L1 Regularization $$R(W) = \sum_{k=1}^{K} |W_k|$$

$$\frac{1}{N}\sum_{i}^{N} Loss_i(f(x_i, W), y_i) + \lambda \boxed{R(W)}$$

Elastic Net - Combination of L1 and L2

Max Norm Regularization, Batch Normalization, Stochastic Depths, Dropout Networks, … many more.

Summary

$$\{(x_i, y_i)\}_{i=1}^{N=3}$$

Summary

$$\{(x_i, y_i)\}_{i=1}^{N=3}$$

```
def predict(image, W):
    return(W*image)
```

Summary

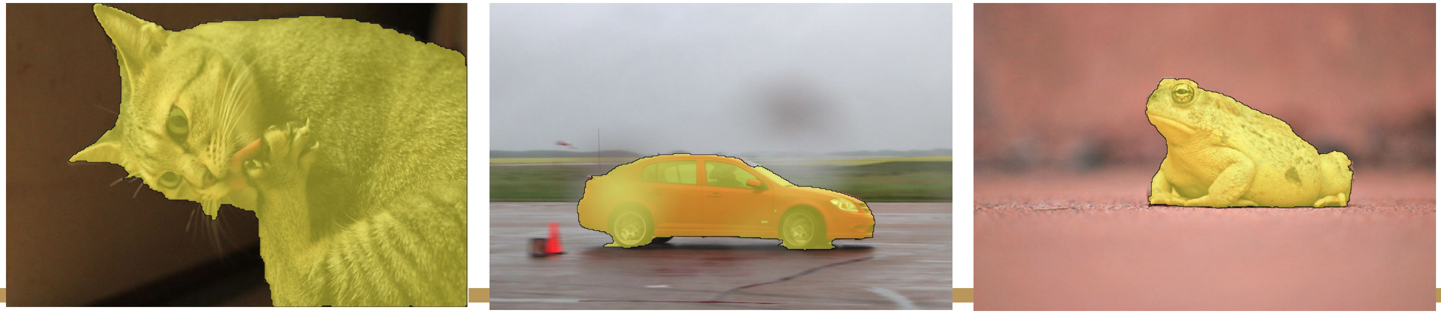
$$\{(x_i, y_i)\}_{i=1}^{N=3}$$

$\downarrow$

```
def predict(image, W):
    return(W*image)
```

$\downarrow$

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1

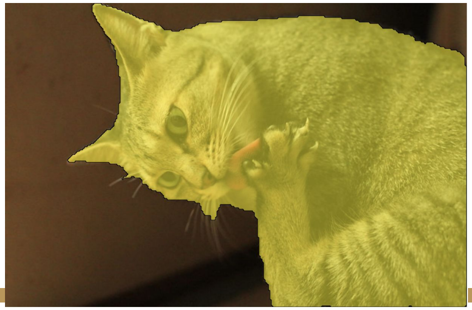
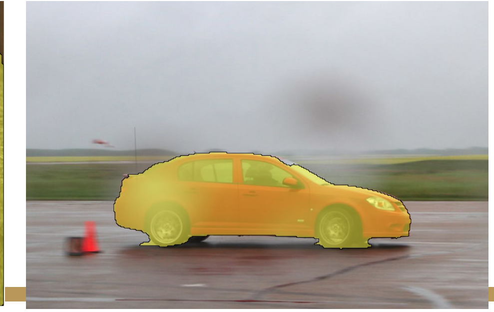


Summary

$$\{(x_i, y_i)\}_{i=1}^{N=3}$$

```
def predict(image, W):
    return(W*image)
```

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1


$$L_i = -log(\frac{e^s_k}{\sum_{j=1}^{J} e^s_j})$$

Summary

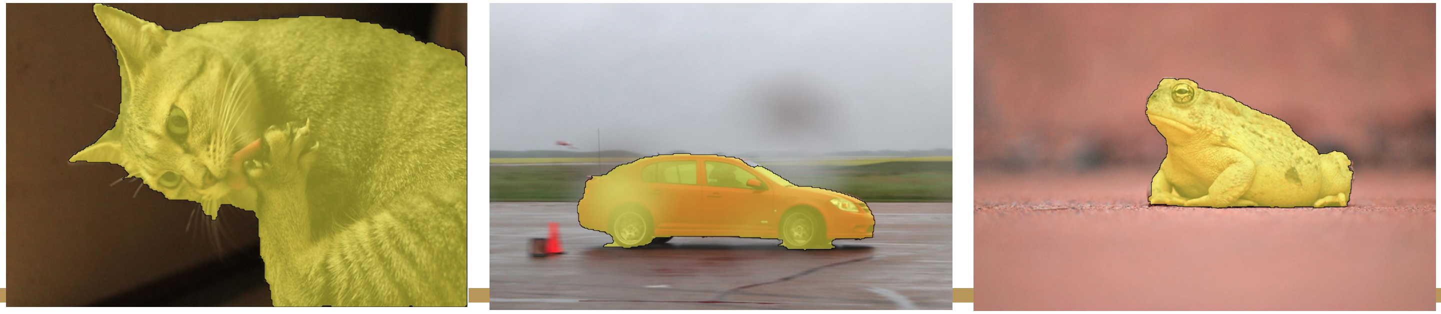
Total Loss=

$$\frac{1}{N}\sum_i^N Loss_i(f(x_i, W), y_i)$$

$$\{(x_i, y_i)\}_{i=1}^{N=3}$$

```
def predict(image, W):
    return(W*image)
```

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1



$$L_i = -log(\frac{e_k^s}{\sum_{j=1}^J e_j^s})$$

Summary

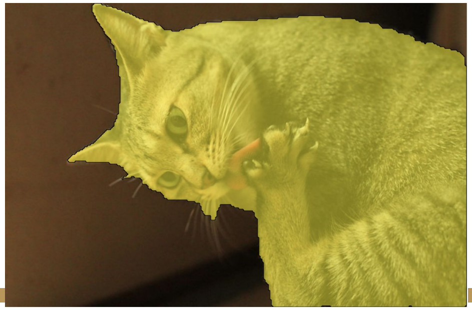
Total Loss=

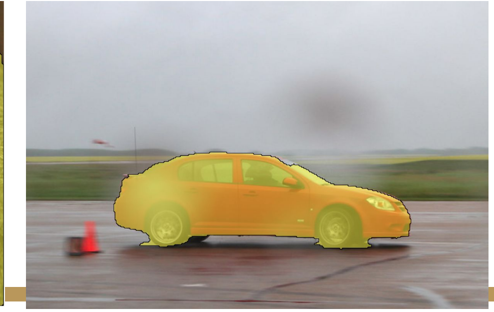
$$\frac{1}{N}\sum_{i}^{N} Loss_i(f(x_i, W), y_i) + \lambda R(W)$$

$$\{(x_i, y_i)\}_{i=1}^{N=3}$$

```
def predict(image, W):
    return(W*image)
```

Cat	3.2	1.3	2.2
Car	5.1	4.9	2.5
Frog	-1.7	2.0	-3.1





$$L_i = -log(\frac{e^s_k}{\sum_{j=1}^{J} e^s_j})$$