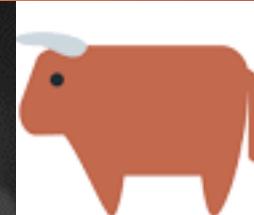


# 3 Minute Challenge

Technical Interview Practice  
for the smart, witty, talented, beautiful human beans  
who fearlessly dissect one AI paper each week:

**OXEN.AI**

PAPER CLUBBERS

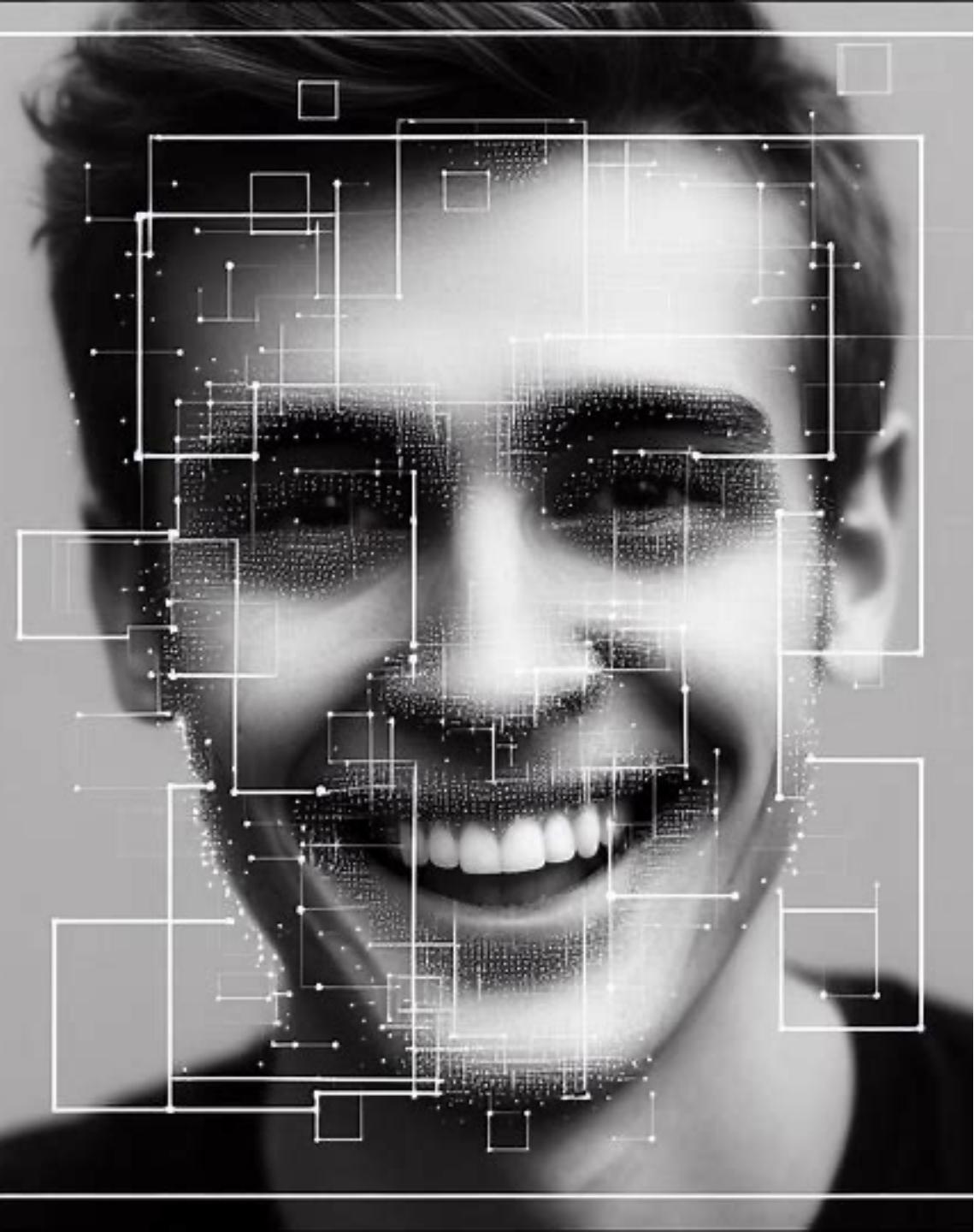


[cameron.github@gmail.com](mailto:cameron.github@gmail.com)

March 24, 2024 [hopper1906](#)



Could you please click hard on my GitHub star button? This CNN is my first thing repo'd.

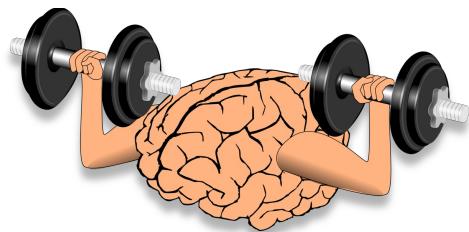


# 3 MINUTE CHALLENGE

**Questions based on**

1. A recent technical interview
2. A verbal 'Stand & Deliver' project interrogation from a MIT Industry Mentor
3. An Austin Deep Learning meet up Q&A

**Foundational Knowledge Topics**



Computer Vision  
Neural Networks

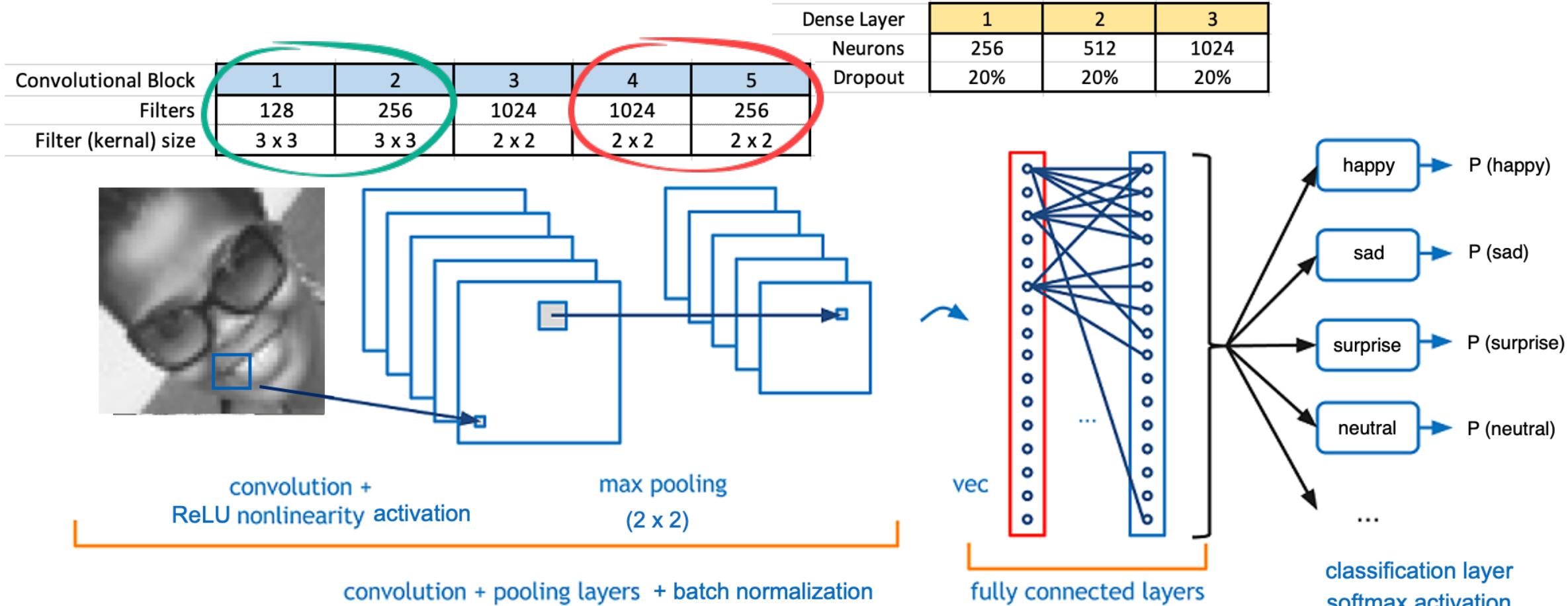
**Only 1 Rule**

Greg  Schoeninger is not allowed to play.

# CONVOLUTIONAL NEURAL NETWORK (CNN)

Facial Emotion Recognition Architecture

What is the difference in types of features that are detected in the very **first** convolutional blocks (ex: 1-2) vs the **final** blocks (ex: 4-5) in any CNN?



# I HAVE THE DATA. I HAVE THE PROOF.

Pick filter sizes carefully.

Good Change  
Bad Change

#	Hyperparameter	From	To	Result on Test Accuracy	Comment
9	Increase Filter size in <b>ALL</b> Convolutional blocks	2 x 2	3 x 3	- 4% .88 --> .84	Each dataset image is only 48 x 48. 3 x 3 is too big a filter size for <b>all</b> 5 conv blocks. Fine detail clues got missed in the end blocks.
10	<b>Increase</b> Filter size <b>ONLY</b> in Convolutional blocks <b>1 - 2</b>	2 x 2	3 x 3	+ 3% .85 --> .88	<b>Early</b> blocks detect big <b>edges</b> and other big <b>details</b> . Keep final 2 convolutional blocks with small 2 x 2 filters that are needed for fine detail detection.

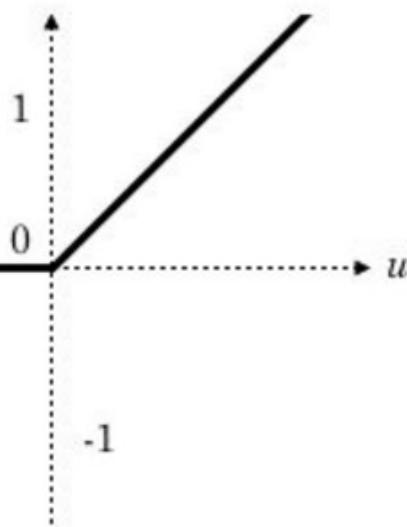
2

# WHAT IS THE PURPOSE OF RELU? HOW DOES IT WORK?

## ACTIVATION IN NEURAL NETWORKS

### My answer

$$f(u) = \max(0, u)$$



Relu (Rectified Linear Unit) is a form of activation. Activation functions add non-linearity to any neural network during both forward and back propagation. Without activation, a neural network would behave like a linear regression.

If the input value is negative, RELU effectively turns off the neuron. If input value is positive, RELU passes it along as-is.

**My answer was assessed as incorrect.**

Why?

# COMPUTER VISION

## 2 FUNDAMENTAL CONCEPTS



2 fundamental concepts make Convolutional Neural Networks (CNNs) effective and efficient in processing images.

- 1) **Spatial Locality** – pixels close to each other in an image are more likely to be related than pixels that are far apart.  
Ex: The pixels that form each Zebra's stripe
- 2) What is the second fundamental concept?  
-----  
(2 words)

# THANK YOU

cameron.github@gmail.com

# APPENDIX

for context  
my Capstone Project presentation  
MIT Data Science and Applied ML program

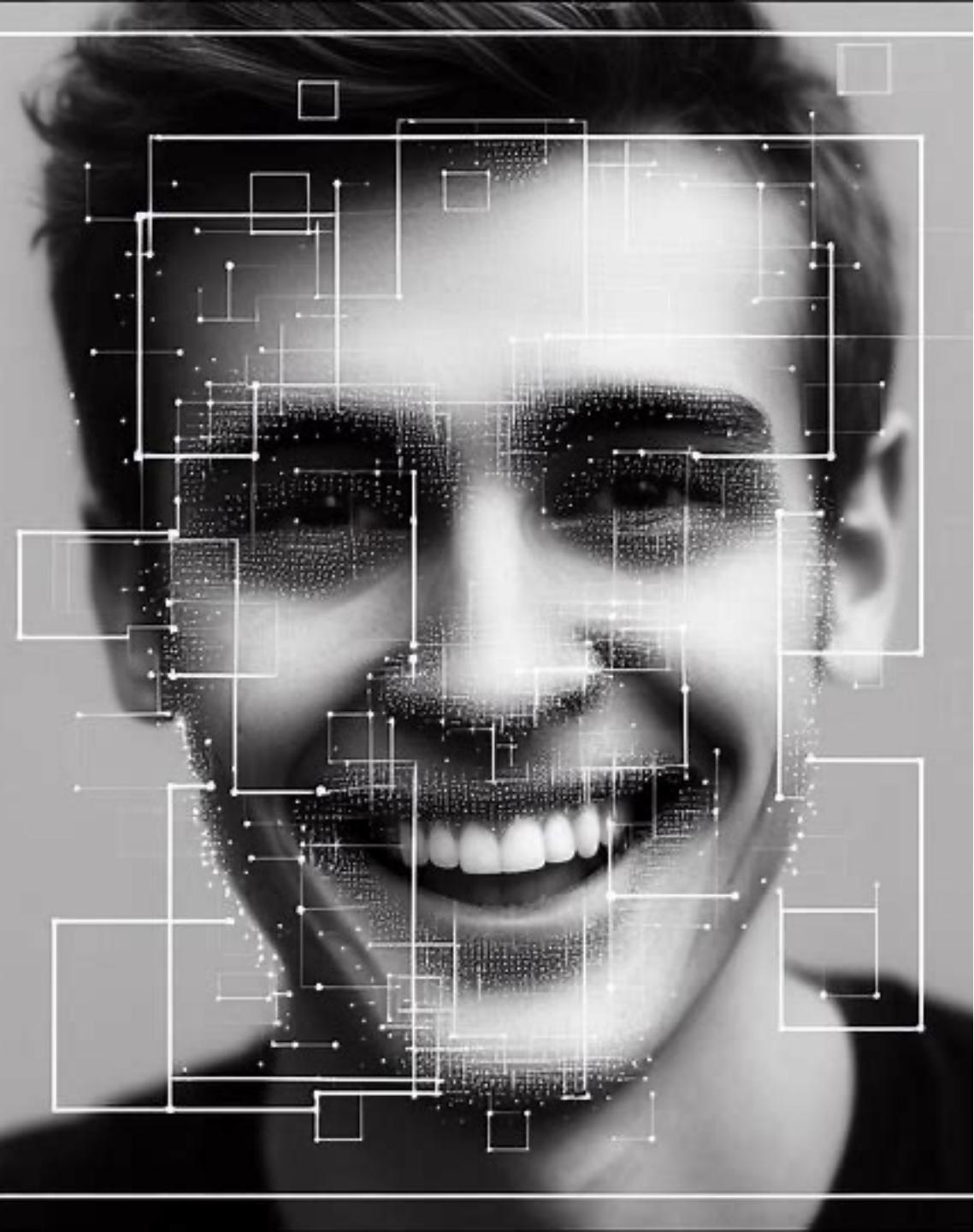
# MoviEmotion

CAPSTONE



cameron.github@gmail.com

March 22, 2024



# AGENDA

MoviEmotion

## Executive Summary

## Business Problem

## Technical Solution

## ROI & What's Next



# MoviEmotion

*HAPPINESS IS IMPORTANT*

## BUSINESS PROBLEM

**Happiness** is important, today more than ever.  
**Entertainment** creates **happiness**.

Movie script writers, both human and AI,  
need to know:

**Which moments** hold highest resonance?  
**Which patterns** of resonant **moments** create the  
best viewer engagement?

The right sequence of resonant moments  
creates the secret recipe to blockbusters and  
sleeper successes.



# EXECUTIVE SUMMARY

MoviEmotion

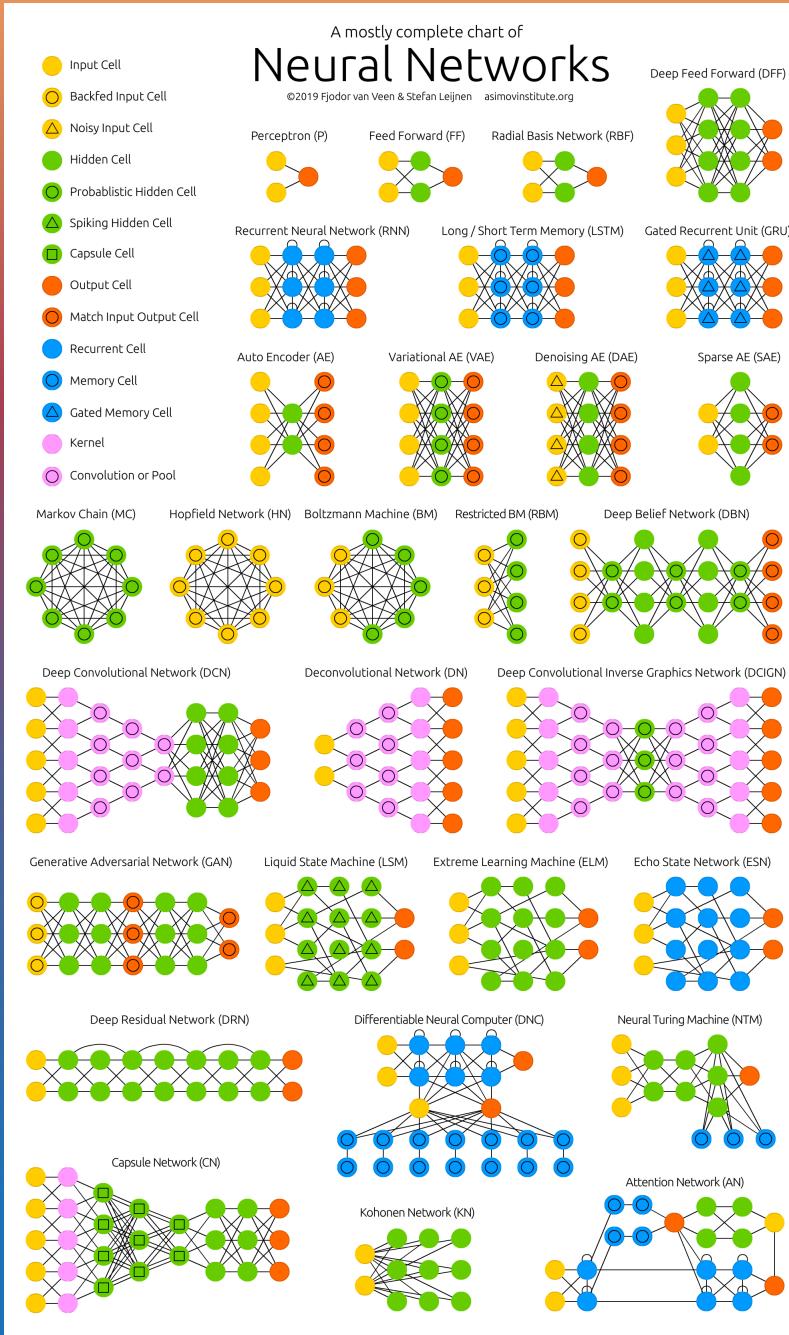
## BUSINESS SOLUTION

[cameron.github@gmail.com](mailto:cameron.github@gmail.com)

- Map a **movie script**, second by second, to the focus group's **timeline of feels**.
- Define the pattern of resonant moments that enable a studio to easily out-Hallmark a Hallmark channel holiday movie.
- **Done!** An emotion-detection neural network model trained on **> 15,000** images to an accuracy of 94%.
  - 36 experiments, aka “training iterations”
  - 40-70 epochs per iteration
  - Each model training iteration took 45 min - 3 days
  - ROI calcs based on Nvidia Tesla A100 GPU and Google Colab Pro+

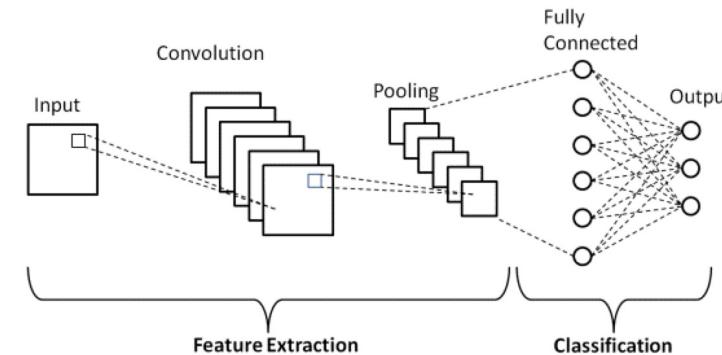


accuracy on  
new test images.



# TECHNICAL SOLUTION

LET'S DIVE IN



A **Convolutional Neural Network (CNN)** is the best neural network architecture for an image recognition problem. ‘Best’ means:

**Highest accuracy. Lowest computational costs**

High accuracy via:

Spatial locality       Translational invariance

Low computational cost via:

Weight sharing between filters

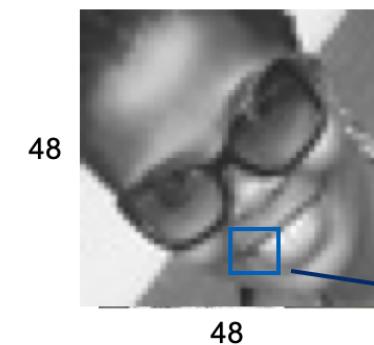
# RECOMMENDATION: CNN MODEL # 36



## MoviEmotion's Emotion Recognition Architecture

cameron.github@gmail.com

	2 x 2 --> 3 x 3 filter size on first two blocks increased accuracy +3%				
	Increasing filter size to 3x3 on later blocks decreased accuracy.				
Convolutional Block	1	2	3	4	5
Filters	128	256	1024	1024	256
Filter (kernel) size	3 x 3	3 x 3	2 x 2	2 x 2	2 x 2



48

convolution +  
ReLU nonlinearity activation

max pooling  
(2 x 2)

ReLU --> Leaky ReLU  
decreased accuracy  
88% to 78%.

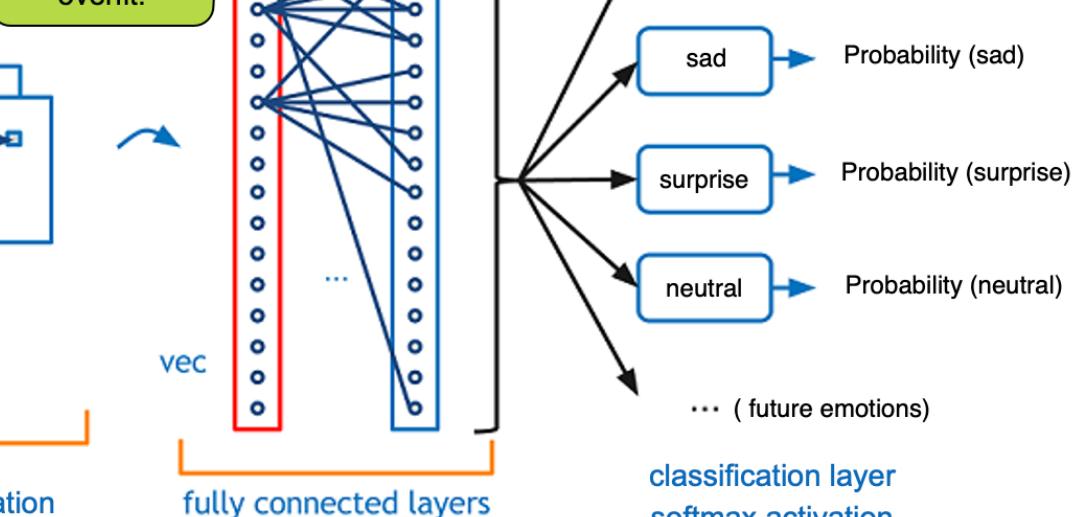
convolution + pooling layers + batch normalization

No observable change

Adding third dense neuron layer  
increased accuracy on new test data  
93 --> 94%

Dense Layer	1	2	3
Neurons	256	512	1024
Dropout	20%	20%	20%

20% = ideal.  
Decreased overfit.



classification layer  
softmax activation

Legend:

Good change

Bad change

# 36 EXPERIMENTS ON MY BIONIC CNN

Good Change

MoviEmotion: Mapping the Timeline of Feels vs a Movie's Script

Bad Change

cameron.github@gmail.com

#	Hyperparameter	From	To	Result on TestAccuracy	Comment
1	Rectified Linear Unit (ReLU) Activation type on all layers <code>model.add(Activation('relu'))</code>	Leaky ReLU	ReLU	+ 10% .78 --> .88	<b>Activation</b> adds non-linearity to the neuron values. Without activation, a neural network would just act like a linear regression model. <b>ReLU activation</b> makes $y = 0$ if $x < 0$ . This means negative values don't get propagated. Leaky ReLU has a tiny negative y slope when x is negative.
2	<b>ReLU</b> on Convolutional blocks <b>Leaky ReLU</b> on Dense Layers	ReLU Dense only	Leaky ReLU Dense only	- 4% .88 --> .84	Some research shows that Leaky ReLU can be better on dense layers. But not for this model!
4	Batch Normalization <b>location</b> BN makes inputs have mean=0, stdev=1	After ReLU	Before ReLU	- 10% .88 --> .78	Watched Ian Goodfellow's YouTube. Decided to experiment with Batch Normalization location.
5	<b>ReduceLROnPlateau</b> <code>min_delta = 0.0001</code>	.0001	.00001	- 3% .88 --> .85	Learning Rate on Plateau is a compile section callbacks list parameter. This slowed the Epochs.
8	Change Optimizer to Stochastic Gradient Descent ( <b>SGD</b> )	Adam	SGD	- 13% .88 --> .75	A neural network's optimizer adjusts weights and learning rates at start of back propagation.
9	Increase Filter size in <b>ALL</b> Convolutional blocks	2 x 2	3 x 3	- 4% .88 --> .84	Each dataset image is only 48 x 48. 3 x 3 is too big a filter size for all 5 conv blocks.
10	<b>Increase</b> Filter size <b>ONLY</b> in Convolutional blocks <b>1 - 2</b>	2 x 2	3 x 3	+ 3% .85 --> .88	That keeps final 2 convolutional blocks with small 2 x 2 filters are needed for fine detail detection.

# 36 EXPERIMENTS ON MY BIONIC CNN

Good Change

MoviEmotion: Mapping the Timeline of Feels vs a Movie's Script (2)

Bad Change

cameron.github@gmail.com

#	Hyperparameter	From	To	Result on TestAccuracy	Comment
12	Slow the Optimizer Learning Rate <code>optimizers.Adam(learning_rate=.0001)</code>	.001	.0001	+ 4% .84 to .88	Optimizer learning rate determines how fast the model moves towards a loss minima. I also increased epochs from 40 to 60.
13	Cut batch size in half <code>model.fit(batch_size = 64)</code>	128	64	- 7% .88 -> .81	This smaller batch size: 1) Provided a worse estimate of the gradient 2) Gave fewer 'clues' to where global loss minima is located.
21	Increase Validation Split <code>model.fit(validation_split=0.3)</code>	20%	30%	+ 3% .88 --> .91	Increasing validation split to 30% helped! Finally: Test accuracy over 90% !
23	Increase Validation Split even more <code>model.fit(validation_split=0.4)</code>	30%	40%	- 10% .91 -> .81	40% validation split: too much. Goldilocks zone: 30%. Not 20. Not 40.
25	Lots of Data Loader Augmentation ex: increase range of brightness parameter	(0.,2.)	(0.7.,2.)	- 19% .91 -> .72	Widening the image brightness level range applied by data loader did NOT work.
27	L2 Regularization <code>model.add(kernel_regularizer=(0.01))</code>	none	0.01	- 35% .91 -> .56	Adding L2 regularization decreased accuracy. -35% (!). Minimal change to fit.
31	Increased neural network dense layer widths	128, 256	256, 512	+ 2% .91 --> .93	Good changes: More neurons per layer. More convolutional filters per block.
*36	Add a third fully connected dense layer <code>model.add(Dense(1024))</code>	2	3	+ 1% .93 --> .94	Adding a third final layer increased test accuracy to 94%. Rah!



# BEST TEST ACCURACY > 36 EXPERIMENTS

PRO+ File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

☰ {x} 🔎

▼ Evaluating the Model on Test Set

```
▶ # model.predict() is a function that generates output predictions on the input samples.  
# By default, it returns the probabilities of each class for the given input.  
# https://saturncloud.io/blog/understanding-keras-modelpredict-returning-classes-instead-of-p  
  
# test_pred = model2.predict(test_set)  
  
# np.argmax() returns the indices of the maximum values along an axis.  
# axis = -1 specifies the last dimension of the array  
# https://numpy.org/doc/stable/reference/generated/numpy.argmax.html  
  
# The last dimension of the array is the one that classifies the emotion.  
  
# test_pred = np.argmax(test_pred, axis = -1)  
  
# Code to evaluate Model CNN_bionic on test data.  
test_images, test_labels = next(test_set)  
accuracy = CNN_bionic_gray.evaluate(test_images, test_labels, verbose = 1)
```

➡ 1/1 [=====] - 0s 39ms/step - loss: 0.1094 - accuracy: 0.9375

Observations on the CNN model 6's - Grayscale accuracy on test data:

- % model accuracy on new test data: 94%



# TRANSFER LEARNING

*"IT SEEMED LIKE A GOOD IDEA"*

cameron.github@gmail.com

Transfer Learning holds the promise of re-using what a pre-trained model already knows. Better accuracy. Less work.

However, 36 experiments later:  
TL model accuracy < my custom model

Why did Transfer Learning **not** work?

- 💣 Color Mode mismatch `rgb vs grayscale`
- 💣 Domain mismatch `object images vs face images`
- 💣 Image dimension mismatch `224x224 vs 48x48`

Ex: Oxford University's Visual Geometry Group (VGG-16) model imported at `transfer_layer = vgg.get_layer('block3_pool')`  
test data accuracy: just 73%

A 3D rendering of a person in a dark suit and tie, seen from the side and back, sweeping a large pile of small white dots (representing binary code) off a wall of computer monitors. The monitors are arranged in a grid and display various numbers and letters, some of which are obscured by the falling dots. The scene is set against a dark background with a blue-to-black gradient.

# DATA QUALITY

CLEAN DATA  
HAPPY SCIENCE

- ▶▶ Next: The next data set must be fully cleaned before model training for the next epic.

Exploratory Data Analysis on > 15,000 images showed:

- Misclassified images
- Corrupted data - Glyphs or solid black / white images

💣 Result: Unhelpful patterns trained my model.

# EXAMPLES – MOSTLY CORRECTLY LABELED DATA

+ Code + Te  
1

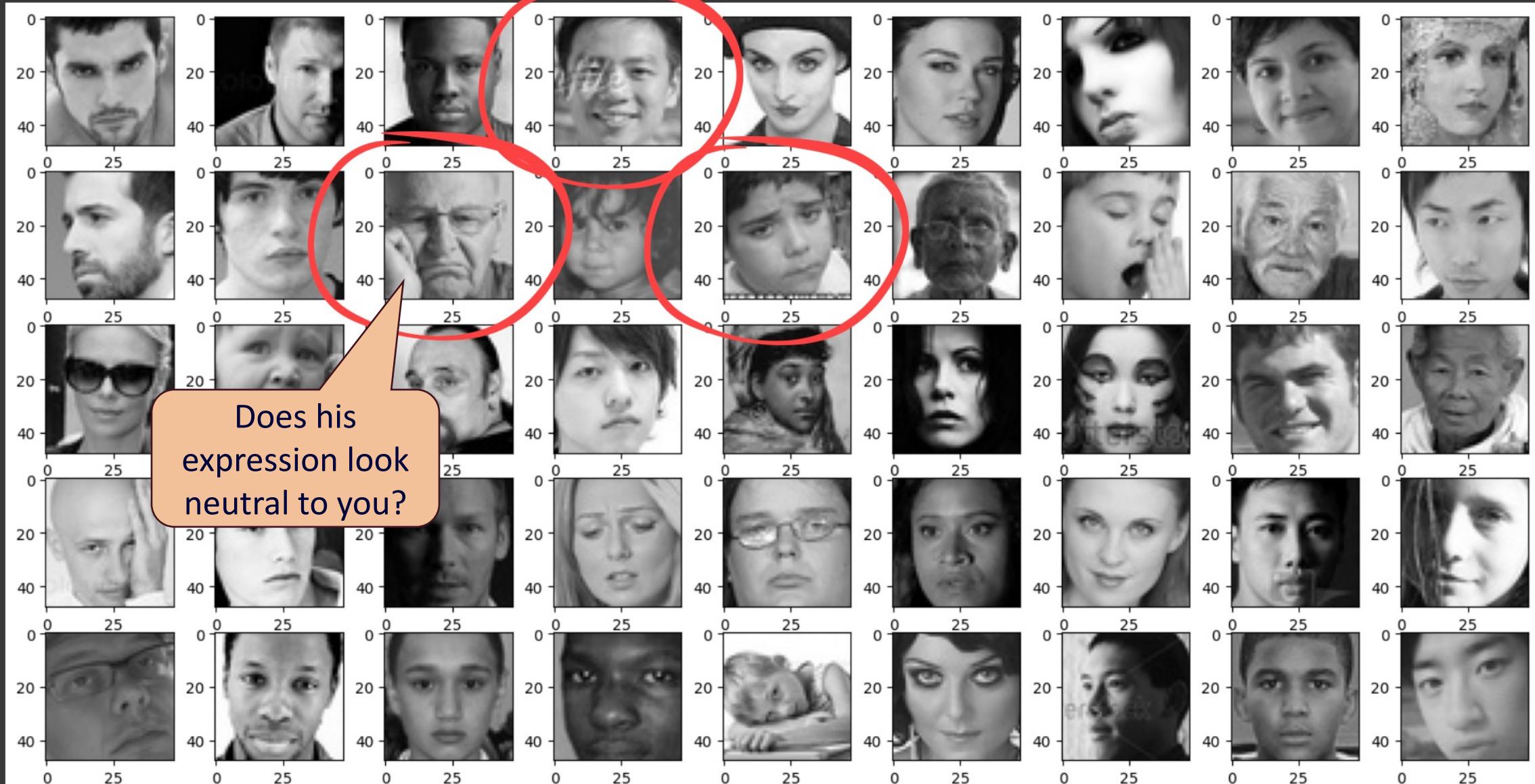
neutral

cameron.github@gmail.com

9

1

5



5

# EXAMPLES – MOSTLY CORRECTLY LABELED DATA

+ Code + Text pit... mg)  
plt.show(),

sad

cameron.github@gmail.com

Connect

1 5 9

1

5

9

This looks like  
Tom Brady's tears of  
happiness after  
a big win. Not sad.

The image displays a 4x10 grid of grayscale face images, each with a bounding box and numerical coordinates (0, 20, 40) on its right side. The images show various people, mostly women, displaying different emotions. A green circle highlights the 5th image from the top row, which depicts a man crying. An orange callout bubble contains the text: "This looks like Tom Brady's tears of happiness after a big win. Not sad." Three other images are circled in red: the 9th image in the 3rd row, the 9th image in the 4th row, and the 9th image in the 5th row. The top of the image features a header bar with 'Code', 'Text', 'pit...', 'mg)', 'plt.show(),', 'sad', 'cameron.github@gmail.com', 'Connect', and navigation icons. On the left, there are large circular numbers 1, 5, and 9. On the right, there are smaller circular numbers 1, 5, and 9.

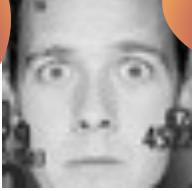
# DATA SHENANIGANS

## INCORRECTLY LABELED DATA

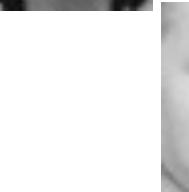
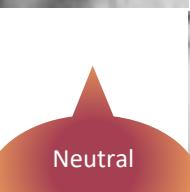
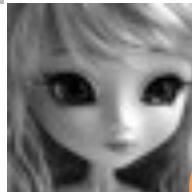
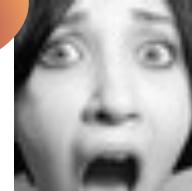
How would **you** label these images?



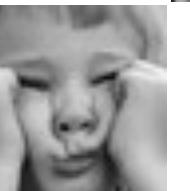
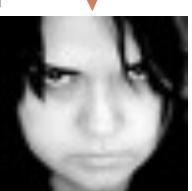
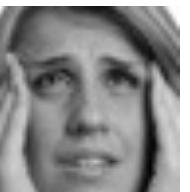
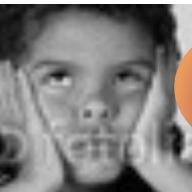
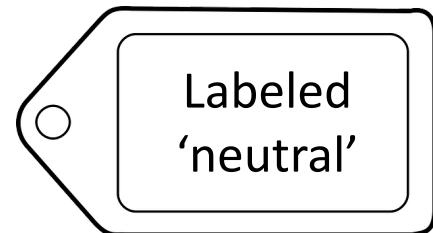
Sad eyes,  
but  
happy  
smile



More  
surprised  
than sad



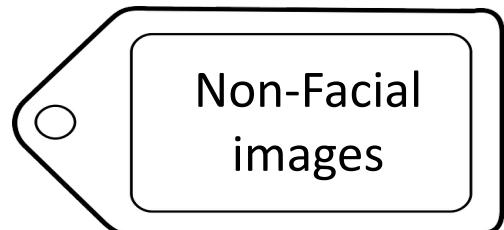
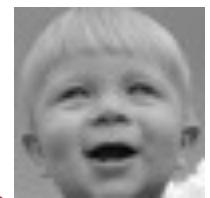
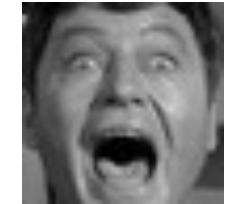
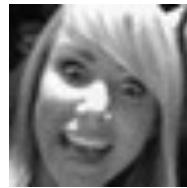
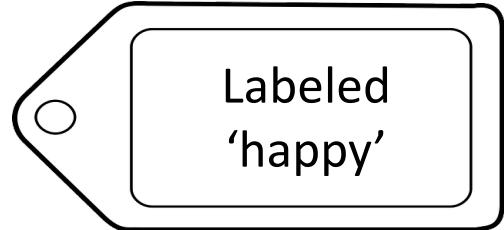
If humans debate which emotion is being felt, how do we expect the machine to learn from it?



# DATA SHENANIGANS

## INCORRECTLY LABELED DATA (2)

How would **you** label these images?



# MACRO OBSERVATIONS

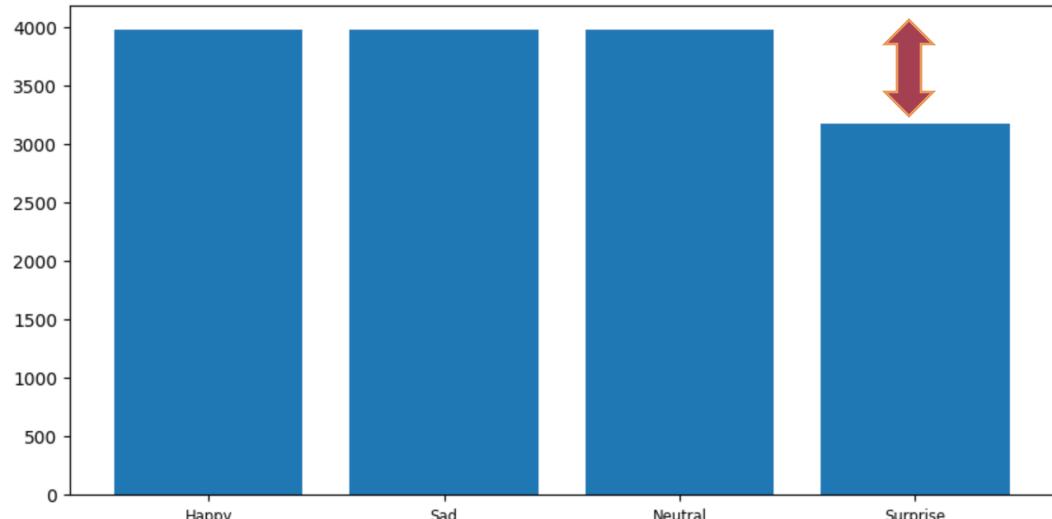
*"EVIDENCE SPEAKS LOUDER THAN INTUITION"* - STATSIG

Exploratory Data Analysis (EDA) also showed:

👁 Fewer surprised images

21% surprised vs 26% of total for each other emotions

**Ideal:** even split of classification category data

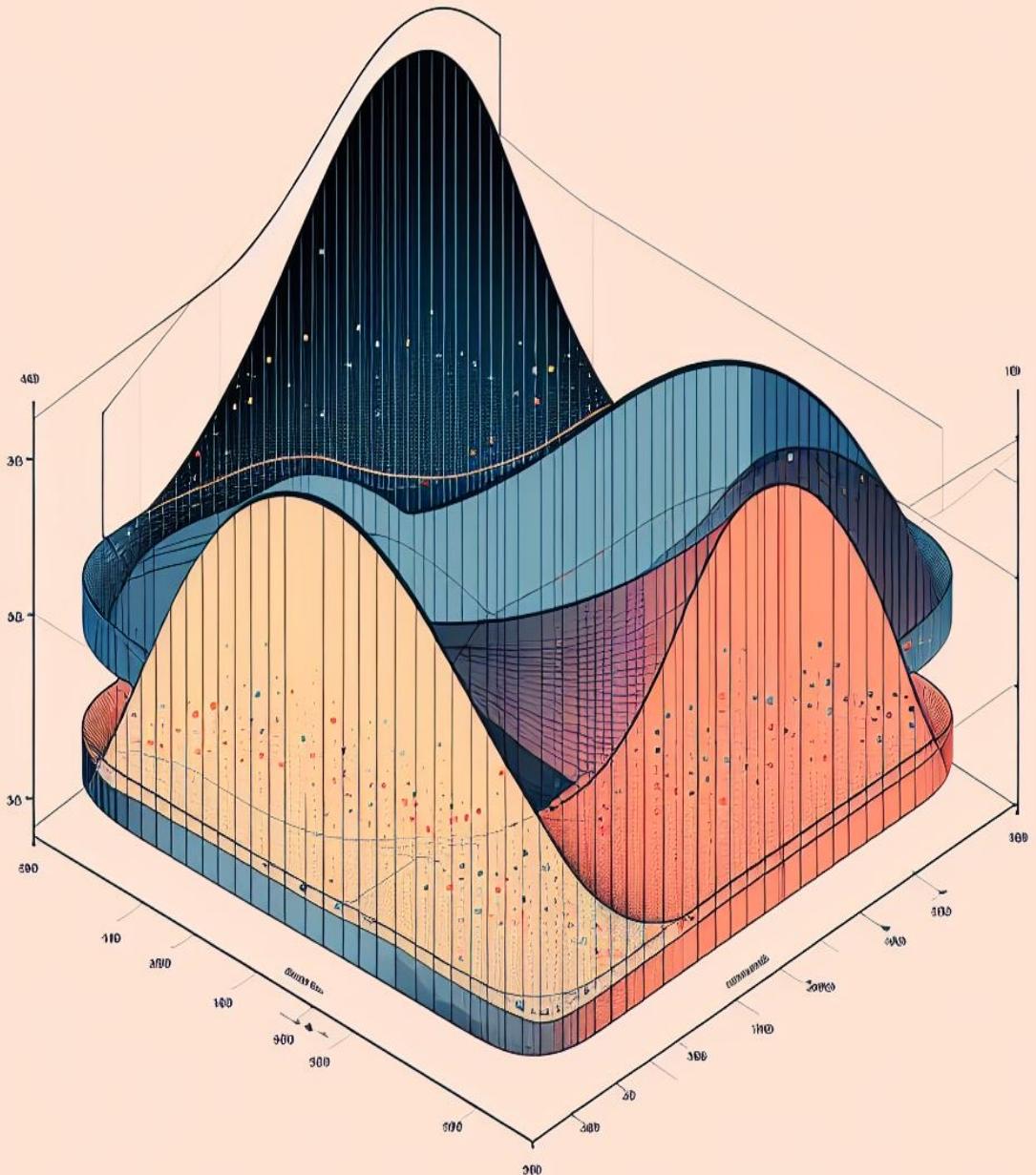


👁 Not enough **test** images.

Test dataset was just 0.6% of total

	Train	Validate	Test
Ideal %	70	20	10

I can address both problems in next sprint.





# RETURN ON INVESTMENT (ROI)

CONSTRAINT: POSITIVE ROI WITHIN 6 QUARTERS

$$\text{ROI} = \frac{\text{Net investment gain}}{\text{Cost of investment}} \times 100$$

I will collaborate with your Financial Analyst to define tangible and intangible types of :

- Benefits
- Costs

for the ROI spreadsheet. Specific focus will be on accuracy of first 6 quarters' cash flow.



## ROI – COSTS

CALCULATED - COMPUTE

Google Cloud Pricing Calculator



COMPUTE  
ENGINE



GKE  
STANDARD



BACKUP FOR  
GKE



CLOUD TPU

WC

Estimate

USD ▾

1,046.77 / mo

Without a fast GPU, model training epochs becomes soggy slow:

Training the model once (60 epochs):

1 Nvidia Tesla A100 GPU



vs

1 standard Google Cloud CPU

**2 DAYS**



# ROI – COSTS – SOFTWARE & COMPUTE

COMPUTE COSTS WILL BE MANAGEABLE

94% accurate Bionic Model #36  
**creation** costs to date:

**\$110: \$100 Google Colab Premium + \$10**

(\$10 additional compute credit purchased after I exceeded the premium subscription GPU time limit)

--> Assume \$5,000 / month first year model costs

- Inference costs from ongoing use will be small vs initial training costs.
- Even ‘huge’ movie focus groups present a **tiny** workload compared other types of business problems. Yay!

Other tangible variable costs to estimate:

Gathering Data, Data Prep, Data Wrangling, Data Analysis, Data Transmission, Model Training, Model Testing, Model Deployment, Model Monitoring

Our friendly Financial Analyst will add to ROI the tangible fixed costs (Salaries, Benefits, Slack, Zoom,...)



## Other **Tangible Costs** (continued):

- Device and installation cost to retrofit focus group screening theater with embedded seat back cameras.
- Legal fees to ensure full disclosure is well done to screening attendees.
- Integration of attendee terms and conditions acceptance into studio's focus group app.

## **Intangible Cost:**

- Opportunity cost of prioritizing this project vs other projects we could do



# ROI - BENEFITS

NEED A GOOD 'CLOSER' ON THE STUDIO CONTRACTS

**Go to Market (GTM)** will target a complete solution service to the top 7 movie studios:

Walt Disney Pictures (\$83 billion), Paramount (30), Lionsgate (36), MGM (13), Universal (12), Sony (7)

## Tangible Benefits

Cash flows from revenue must 10x the final burdened tangible costs over a contract period of "\$ \_\_\_\_ per season of movies"

## Intangible Benefits

- Experience with implementing a verticalized AI application involving both hw and sw
- Brand awareness & goodwill via media attention



# WHAT'S NEXT: PRIVACY

LET'S NOT BE CREEPY

- Complete, transparent, easy to skim disclosure.
- No personally identifiable information added to the image
- Image deletion after a defined period

## RISKS

BIGGEST UNKNOWNS

- Focus group theater seat camera installation costs. Need studios to cover.
- Could we better position this solution? Have we figured out the studios' biggest pain point we can solve with it?

A photograph of a sunset over a mountain range. The sun is low on the horizon, casting bright rays of light across the sky and illuminating the peaks of the mountains. The foreground shows a dark, rocky slope. The sky is a clear blue at the top, transitioning to a warm orange and yellow near the horizon.

# WHAT'S NEXT

WHAT'S THE BIGGEST PAIN TO BE SOLVED?

- Listen hard to movie studio first impression : What is the **biggest insight** this can reveal for them? Audience sentiment flow and pattern? Or something else?  
Could **Trailer reaction patterns** be predictive? **Usefully** predictive? How?
- Consider defining a per-person “**neutral**” **baseline** from which to measure emotion. One person’s “mildly surprised” is another person’s “shocked”.
- Add **strength** of emotions
- Add more emotion **categories**. “tears of joy”, “happily surprised”

# CREDITS

## PEOPLE WHO INSPIRED

**MIT Professors** Munther Dahleh, Stefanie Jegelka, Devavrat Shah, John Tsitsiklis, Caroline Uhler

**NYU Professor** Trush Majmudar

**Learning  
Mentors,  
Instructors,  
SMEs** Cristiano Aguiar, Benjamin Artuso, Jacob Curtis, Dmitri, Matthew Graziano, Scott Howard, Georg Huettenegger, Sayed Mohammad Parvasi, Fred Premji, Sanat, Abhinanda Sarkar, Bhaskarjit Sarmah, Greg Schoeninger, D. Sculley, Akshey Singhal, Bradford Tuckfield, Davood Wadi

**Colleagues** Josh Bode, Nathaniel Brewer, William Cole, Nadiya Dragneva, Ameche Egbe, Rama Garg, Mike Gerges, Yvan Giroud, Stephen Lenzen, Aniruddha Mandrill, Chris Marciak, Zara Mubeen, Jennifer Northrup, Evan Pierce, Rama Rao, Joseba Ruiz, Tarachand Sahoo, Harp Singh, William Stanislaus, Jeff Stearns, Eliana Wassermann, Shyam Yadati

**Friends** Matt Kenney, Mike Lake

# CREDITS

## CREATORS OF LEARNING



- **3 Blue 1 Brown**  
(excellent YouTube)
- Arxiv
- Bard
- Bing Image Creator  
(Most images created by Bing)
- Dall-e Image Creator, OpenAI
- Data Base Camp (German)
- Data Bricks
- **Geeks For Geeks**  
(best Python quick reference)
- **GitHub**
- IIETA
- Jcchouinard
- **Kaggle**
- Learn Data Sci
- **Machine Learning Mastery**
- MDPI
- **Medium**
- **Neptune.ai**
- Neural Network Zoo
- **OXEN.AI**
- **Phind** (Killer helpful tool. Thank you Bhaskarjit Sarmah for that tip!)
- Real Python
- Saturn Cloud AI
- Springer
- Stack Exchange
- Stack Overflow
- **StatQuest** (excellent YouTube)
- TensorFlow
- **Towards Data Science**

Photo: Isaac Asimov

cameron.github@gmail.com

Thank  
you!