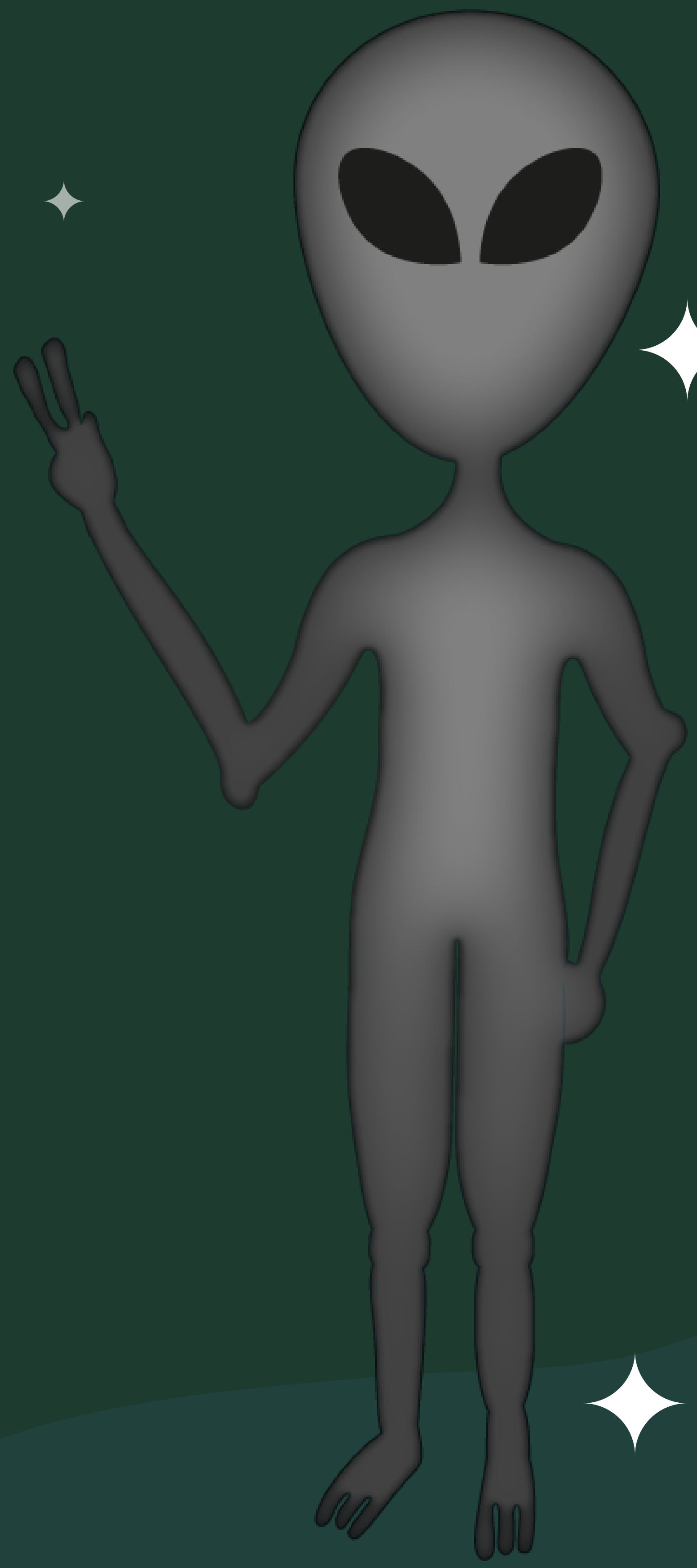
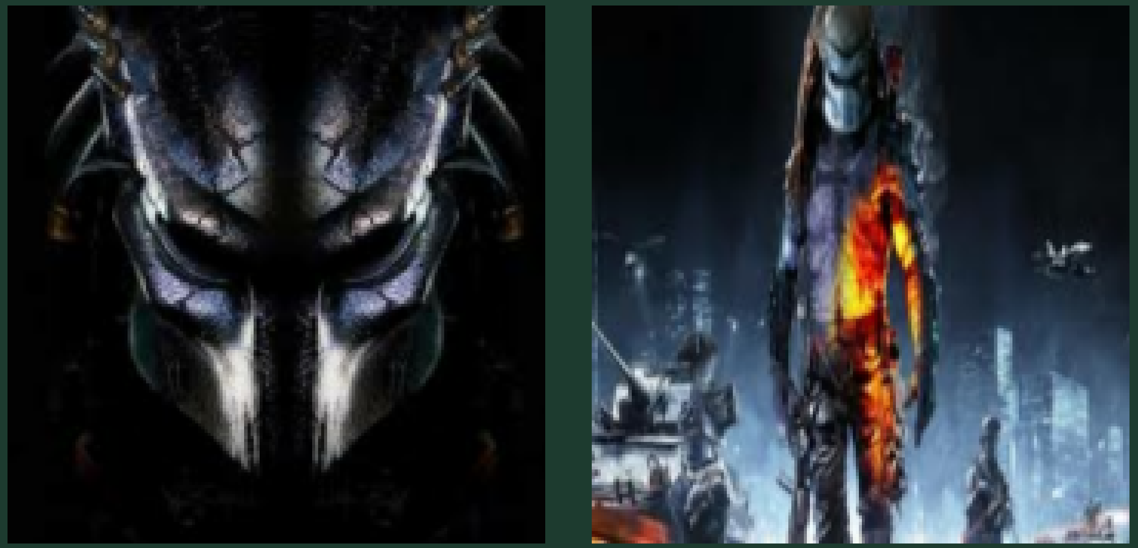
ALIEN VS PREDATOR CLASSIFICATION



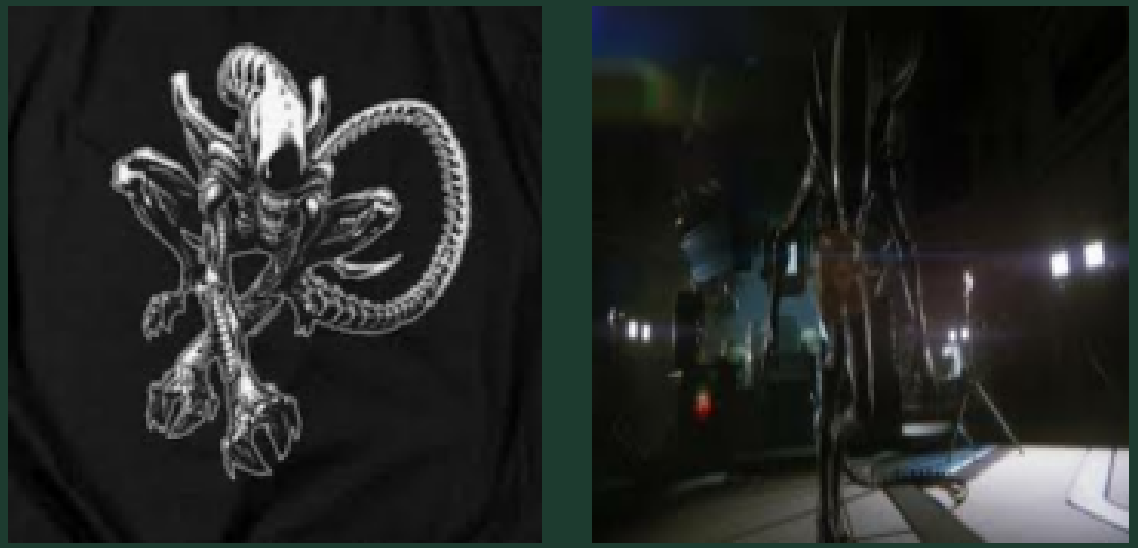
Michalina Hulak

INTRODUCTION

The Alien vs Predator project is a computer vision project that aims to classify images of two popular sci-fi characters, Alien and Predator. The project uses deep learning techniques to build an image classification model that can accurately identify whether an image contains an Alien or a Predator.



predator



alien

Aliens and Predators are two iconic extraterrestrial species from the science-fiction franchise. Aliens are known for their aggressive behavior, exoskeleton body, and the ability to reproduce quickly, while Predators are characterized by their advanced technology, hunting skills, and honor code. Aliens are often depicted as mindless killing machines, while Predators are portrayed as skilled warriors who hunt for sport. Overall, Aliens and Predators are two very distinct and unique species that have captured the imagination of fans for decades.

FOR BUSINESS

The Alien vs Predator project has several applications, including entertainment, education, and marketing. From an entertainment perspective, the project can be used to develop video games or interactive media that involve Alien and Predator characters. In education, the project can be used to teach computer vision and deep learning concepts to students.

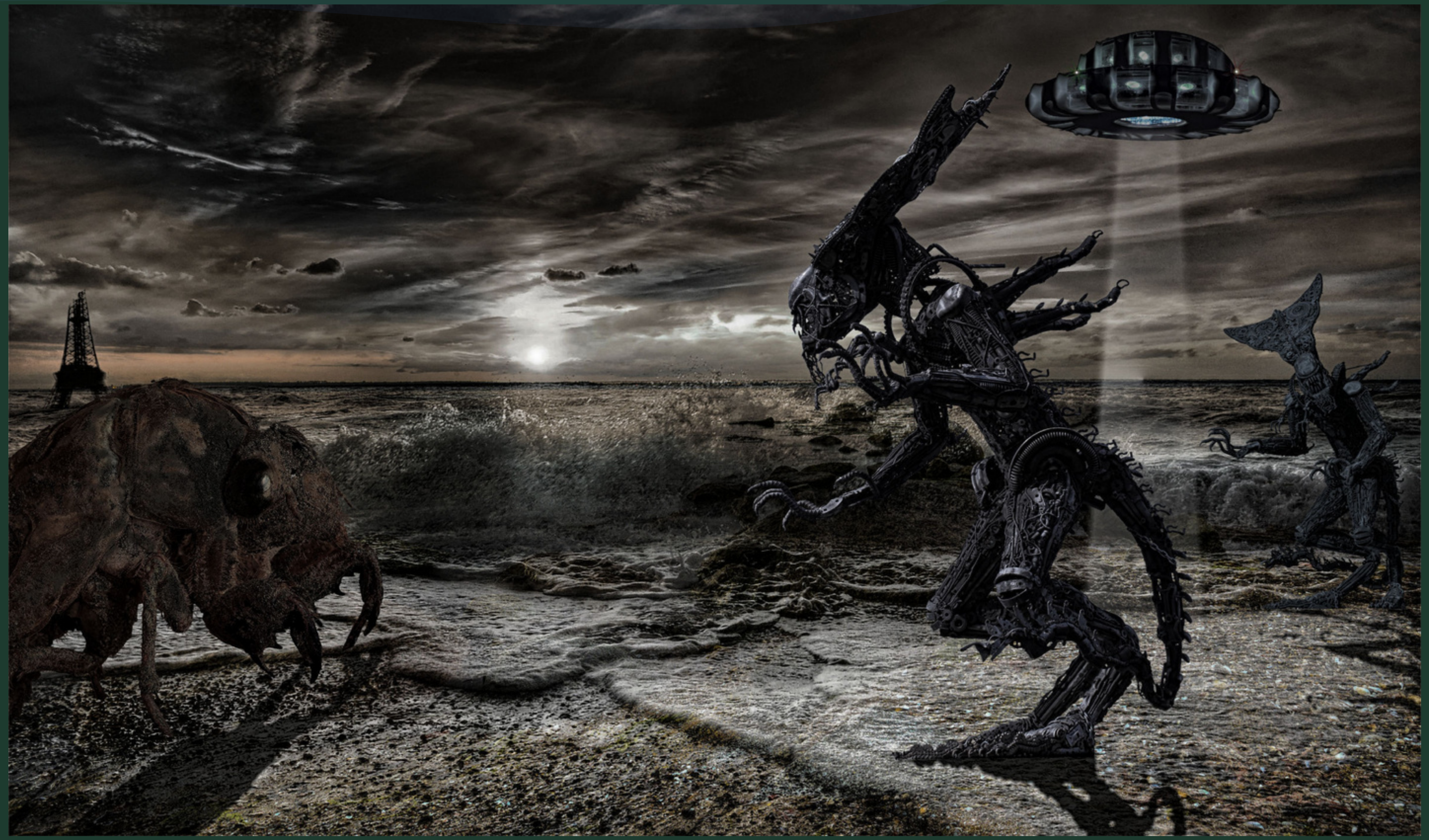
Moreover, from a marketing perspective, the Alien vs Predator project can be used to analyze customer data and preferences by collecting images of customers' favorite character. Companies can use this data to develop targeted marketing campaigns or even personalized products that align with customer preferences.

In conclusion, the Alien vs Predator project is not only a fun computer vision project but also has practical business applications. By building an accurate image classification model, the project can provide valuable insights into customer preferences and drive business growth.

GOAL

The goal is to build a model that can accurately differentiate between the two classes.

I used a deep learning approach, specifically a pre-trained ResNet-50 model, to perform image classification. The dataset consists of images of aliens and predators, and I divided it into training, validation, and test sets. I also used Gradio to create a simple web application that allows users to upload their own images and received a prediction on whether the image contains an alien or a predator.



MODELS

DEEP-LEARNING MODELS KERAS

```
data_augmentation = tf.keras.Sequential([
  RandomFlip("horizontal"),
  RandomRotation(0.2),
  RandomZoom(0.2),
])
```

```
model1 = tf.keras.Sequential([
  tf.keras.layers.experimental.preprocessing.Rescaling(1./255,
                                                        input_shape=input_shape),
  data_augmentation,
  tf.keras.layers.Conv2D(32, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Conv2D(64, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Conv2D(128, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(num_classes, activation='softmax')
])
```

```
model2 = tf.keras.Sequential([
  tf.keras.layers.experimental.preprocessing.Rescaling(1./255,
                                                        input_shape=input_shape),
  data_augmentation,
  tf.keras.layers.Conv2D(32, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Conv2D(64, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Conv2D(128, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(num_classes, activation='softmax')
])
```

```
model3 = tf.keras.Sequential([
  tf.keras.layers.experimental.preprocessing.Rescaling(1./255,
                                                        input_shape=input_shape),
  tf.keras.layers.Conv2D(64, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Conv2D(128, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Conv2D(256, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(256, activation='relu'),
  tf.keras.layers.Dense(num_classes, activation='softmax')
])
```

```
model4 = tf.keras.Sequential([
  tf.keras.layers.experimental.preprocessing.Rescaling(1./255,
                                                        input_shape=input_shape),
  data_augmentation,
  tf.keras.layers.Conv2D(64, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Conv2D(128, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Conv2D(256, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(256, activation='relu'),
  tf.keras.layers.Dense(num_classes, activation='softmax')
])
```

PRE-TRAINED RESNET-50 MODEL

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 7, 7, 2048)        23587712

 global_average_pooling2d (G  (None, 2048)             0
 lobalAveragePooling2D)

 dense (Dense)               (None, 2)                 4098

=================================================================
Total params: 23,591,810
Trainable params: 4,098
Non-trainable params: 23,587,712
```

RESULTS

DEEP-LEARNING MODELS KERAS

	Model_name	train_acc	train_loss	val_acc	val_loss
0	Model1	0.672911	0.645593	0.610	0.741953
1	Model2	0.746398	0.538414	0.720	0.557227
2	Model3	0.942363	0.149516	0.715	0.807535
3	Model4	0.727666	0.543392	0.690	0.588158

PRE-TRAINED RESNET-50 MODEL

train_acc	train_loss	val_acc	val_loss	test_acc	test_loss
0.967797	0.090449	0.923077	0.184397	0.935	0.177922

RESULTS

Looking at the results of the models, we can see that:

- Model 2 and Model 4 both have relatively similar training and validation accuracies, but Model 1 has a lower validation loss, indicating better generalization performance. Model 1 has lower training and validation accuracies compared to Model 2 and 4, and a higher validation loss, indicating it may not perform as well on unseen data. Additionally, the addition of data augmentation to the models can help to improve performance, as seen in Model 4.

- Model 3 has a significantly higher training accuracy compared to the other models, but a much lower validation accuracy and a very high validation loss, indicating it is overfitting to the training data.

- The pre-trained ResNet50 model performs very well with a high training accuracy, high validation accuracy, and low validation loss. The test accuracy is also high, indicating good performance on unseen data.

- Overall, the ResNet50 model seems to perform the best out of the models tested.