Visual Programming: Compositional visual reasoning without training

Qualitative Results

This deck shows numerous successful visual rationales generated by VisProg for the following tasks

- ➢ Compositional Visual QA (GQA)
- ➢ Reasoning with image pairs (NLVR)
- ➢ Factual Knowledge Object Tagging
- ➢ Image Editing with Natural Language

We end with failure cases which are mainly caused by

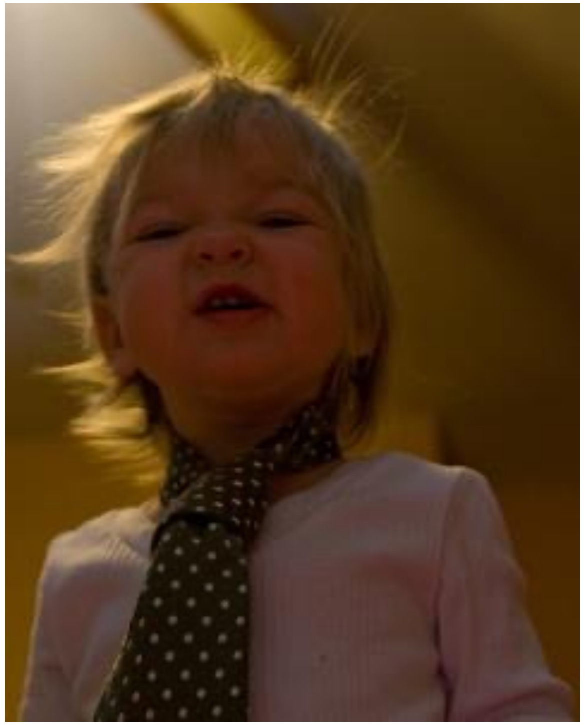
- ➢ Logically incorrect programs
- ➢ Incorrect prediction from a module

GQA

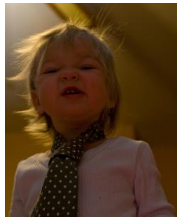
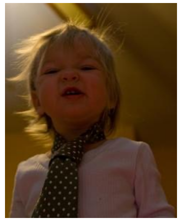
✓ Answer compositional visual questions

Compositional Question Answering

Are there both ties and glasses in the picture?

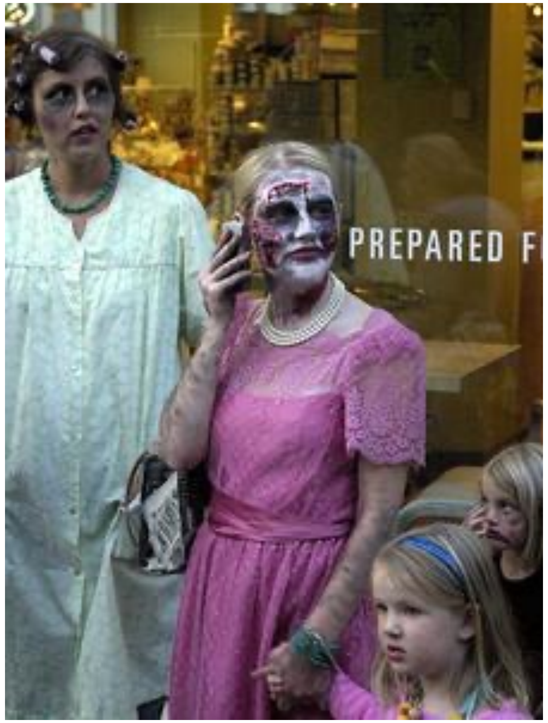


Prediction: no

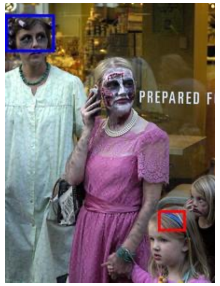
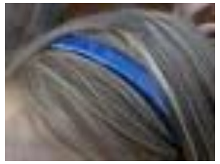
	IMAGE
	BOX0=Loc(image=IMAGE, object=‘ties’)
1	ANSWER0=Count(bbox=BOX0)
	BOX1=Loc(image=IMAGE, object=‘glasses’)
0	ANSWER1=Count(bbox=BOX1)
no	ANSWER2=Eval(expr="'yes' if {ANSWER0}>0 and {ANSWER1}>0 else 'no’”) =Eval(expr="'yes' if 1>0 and 0>0 else 'no'")

Compositional Question Answering

Is the head band brown or blue?

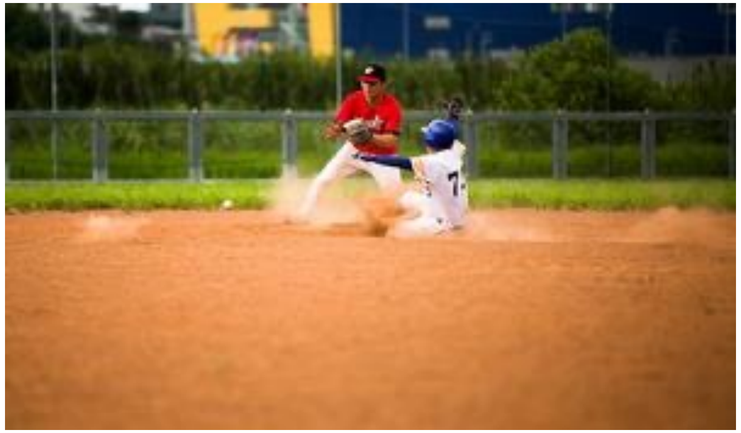


Prediction: blue

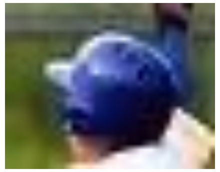
	IMAGE
	BOX0=Loc(image=IMAGE, object=‘head band’) For GQA, we visualize the highest-ranking box in red. The remaining in blue.
	IMAGE0=Crop(bbox=BOX0) When more than 1 boxes are provided to Crop, only the first (i.e., the highest-ranking) box is cropped
blue	ANSWER0=Vqa(image=IMAGE0, question=‘What color is the head band?’)

Compositional Question Answering

Is there a helmet in the photo that is not blue?

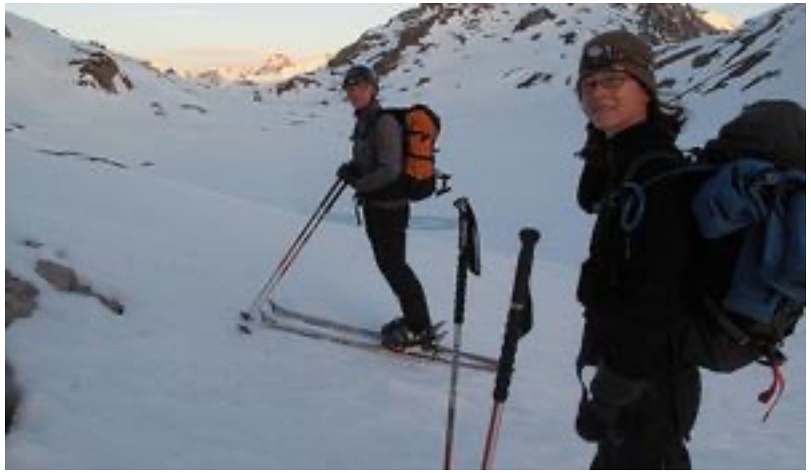


Prediction: no

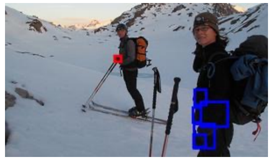
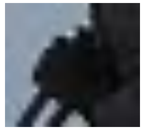
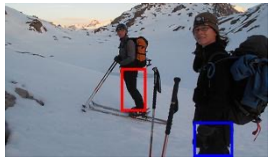
	IMAGE
	BOX0=Loc(image=IMAGE, object=‘helmet’)
	IMAGE0=Crop(bbox=BOX0)
blue	ANSWER0=Vqa(image=IMAGE0, question=‘What color is the helmet?’)
no	ANSWER1=Eval(expr="'yes' if {ANSWER0} != 'blue' else 'no'") =Eval(expr="'yes' if 'blue' != 'blue' else 'no'")

Compositional Question Answering

Do the gloves and the pants have the same color?



Prediction: no

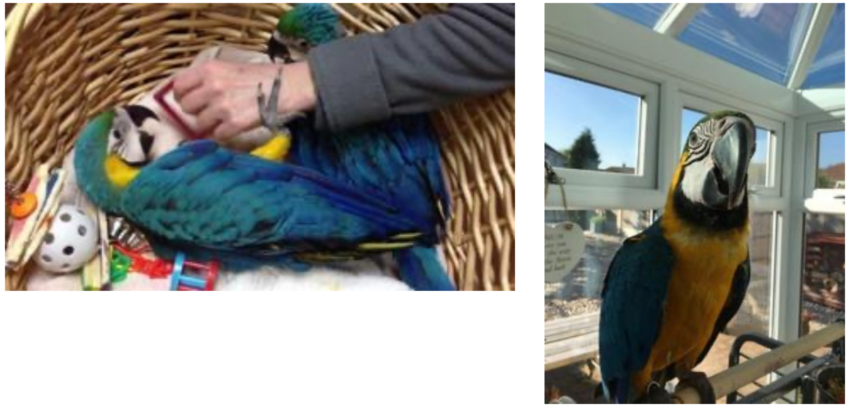
	IMAGE
	BOX0=Loc(image=IMAGE, object=‘gloves’)
	IMAGE0=Crop(bbox=BOX0)
	BOX1=Loc(image=IMAGE, object=‘pants’)
	IMAGE1=Crop(bbox=BOX1)
black	ANSWER0=Vqa(image=IMAGE0, question=‘What color are the gloves?’)
black	ANSWER1=Vqa(image=IMAGE1, question=‘What color are the pants?’)
yes	ANSWER2=Eval(expr="'yes' if {ANSWER0}=={ANSWER1} else 'no'") =Eval(expr="'yes' if 'black'=='black' else 'no'")

Natural Language Visual Reasoning (NLVR)

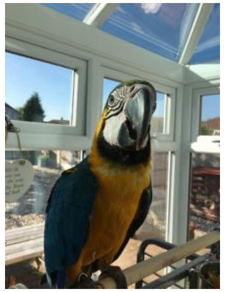
✓ Verify statements about image pairs using a single-image VQA model

NLVR

In one image, a blue and yellow bird is grasping onto a human hand with its talon.

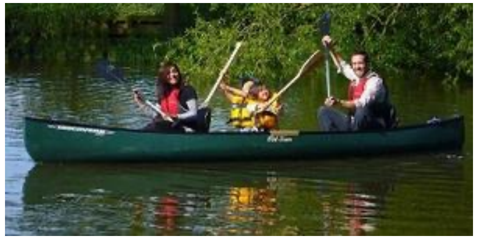


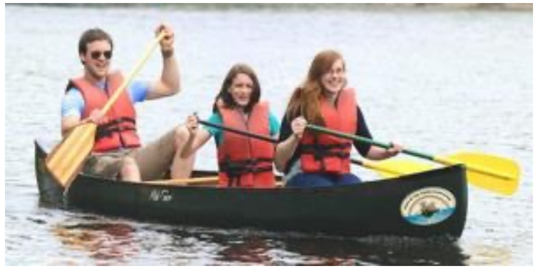
Prediction: True

	LEFT
	RIGHT
yes	ANSWER0=Vqa(image=LEFT, question=‘Is there a blue and yellow bird in the image?’)
no	ANSWER1=Vqa(image=RIGHT, question=‘Is there a blue and yellow bird in the image?’)
yes	ANSWER2=Vqa(image=LEFT, question=‘Is the bird grasping onto a human hand with its talons?’)
no	ANSWER3=Vqa(image=RIGHT, question=‘Is the bird grasping onto a human hand with its talons?’)
True	ANSWER4=Eval(expr="ANSWER0 and ANSWER2") =Eval(expr="True and True")
False	ANSWER5=Eval(expr="ANSWER1 and ANSWER3") =Eval(expr="True and False")
True	ANSWER6=Eval(expr="ANSWER4 xor ANSWER5") =Eval(expr="True != False")

NLVR

The left and right image contains a total of six people and two boats



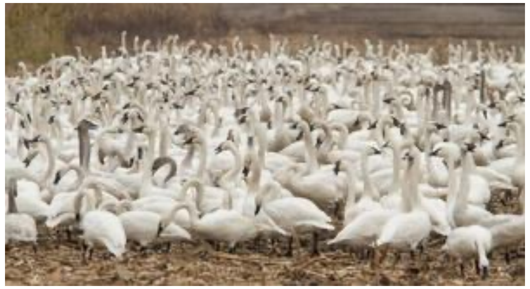


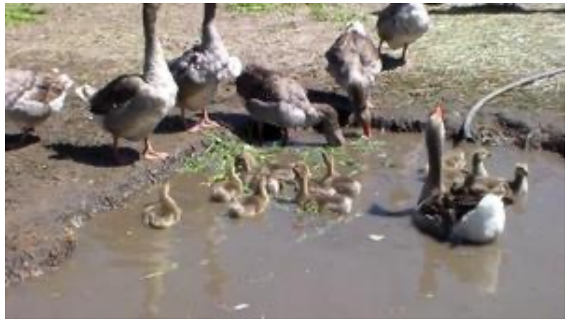
	LEFT
	RIGHT
4	ANSWER0=Vqa(image=LEFT, question=‘How many people are in the image?’)
3	ANSWER1=Vqa(image=RIGHT, question=‘How many people are in the image?’)
1	ANSWER2=Vqa(image=LEFT, question=‘How many boats are in the image?’)
1	ANSWER3=Vqa(image=RIGHT, question=‘How many boats are in the image?’)
False	ANSWER4=Eval(expr="ANSWER0+ANSWER1==6 and ANSWER2+ANSWER3==2") =Eval(expr="4 + 3 == 6 and 1 + 1 == 2")

Prediction: False

NLVR

There are more birds in the image on the right than in the image on the left.



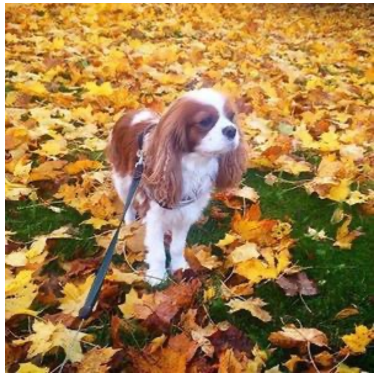


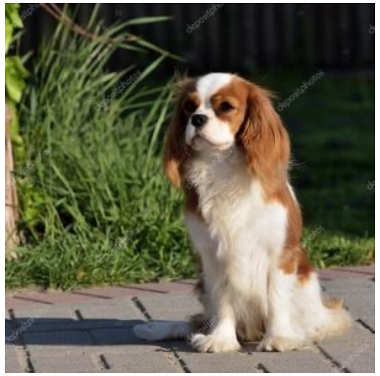
Prediction: False

	LEFT
	RIGHT
100	ANSWER0=Vqa(image=LEFT, question=‘How many birds are in the image?’)
6	ANSWER1=Vqa(image=RIGHT, question=‘How many birds are in the image?’)
False	ANSWER4=Eval(expr="{ANSWER0} < {ANSWER1}") =Eval(expr="100 < 6")

NLVR

A white and brown dog is standing in fallen leaves.





Prediction: True

	LEFT
	RIGHT
yes	ANSWER0=Vqa(image=LEFT, question=‘Is the dog white and brown?’)
no	ANSWER1=Vqa(image=RIGHT, question=‘Is the dog white and brown?’)
yes	ANSWER2=Vqa(image=LEFT, question=‘Is the dog standing in fallen leaves?’)
no	ANSWER3=Vqa(image=RIGHT, question=‘Is the dog standing in fallen leaves?’)
True	ANSWER4=Eval(expr="ANSWER0 and ANSWER2") =Eval(expr="True and True")
False	ANSWER5=Eval(expr="ANSWER1 and ANSWER3") =Eval(expr="True and False")
True	ANSWER6=Eval(expr="ANSWER4 xor ANSWER5") =Eval(expr="True != False")

Factual Knowledge
Object Tagging

- ✓ Tag Personalities
- ✓ Tag Brands & Logos
- ✓ Tag Objects

Tag Personalities

Tag these prime ministers of North American and Asian countries

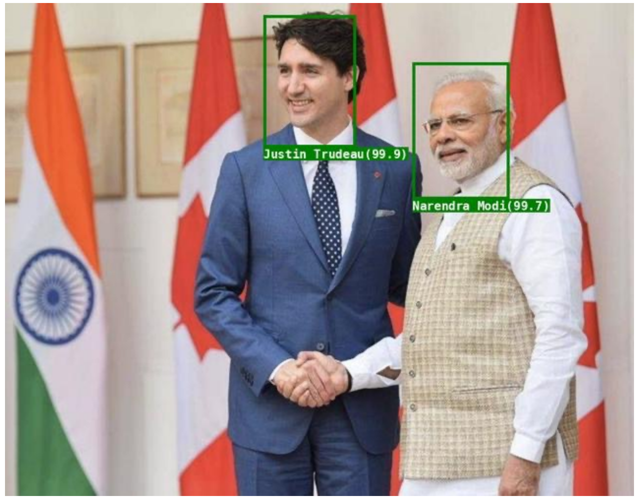


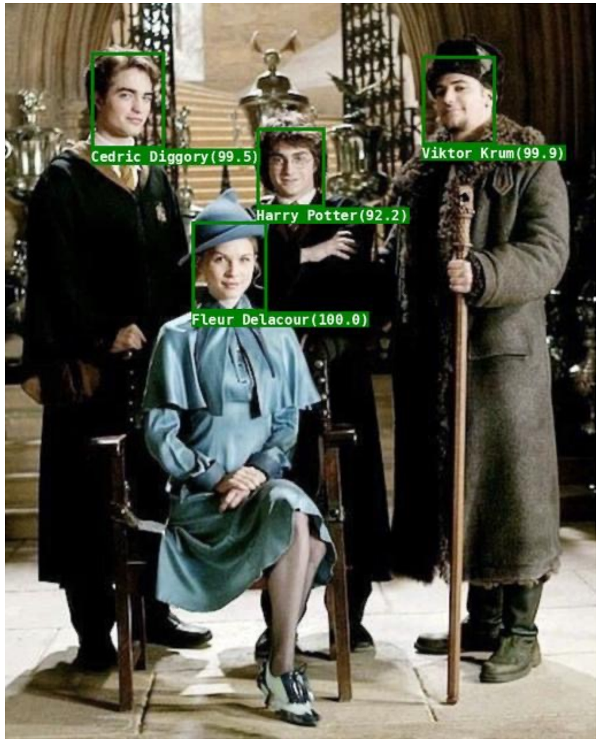
	IMAGE
	OBJ0=FaceDet(image=IMAGE)

['Justin Trudeau', 'Shinzo Abe', 'Narendra Modi', 'Malcolm Turnbull', 'Bill English', 'Nguyen Xuan Phuc', 'Hun Sen', 'Jacinda Ardern', 'Moon Jae-in', 'Tsai Ing-wen', 'Rodrigo Duterte', 'Sheikh Hasina', 'Mahathir Mohamad', 'Prayut Chan-o-cha', 'Joko Widodo', 'Khaleda Zia', 'Giorgi Kvirikashvili', 'Najib Razak', 'Hun Sen']	LIST0=List(
	query=‘prime ministers of North American and Asian countries’,
	max=20)

[{'box': [544, 79, 673, 261], 'tag': 'Narendra Modi', 'score': 99.7}, {'box': [345, 16, 468, 192], 'tag': 'Justin Trudeau', 'score': 99.9}]	OBJ1=Classify(image=IMAGE, categories=LIST0)
	IMAGE0=Tag(image=IMAGE, objects=OBJ1)

Tag Personalities

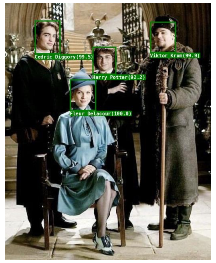
Tag these 4 Triwizard Tournament champions



	`IMAGE`
	`OBJ0=FaceDet(image=IMAGE)`

['Harry Potter', 'Cedric Diggory', 'Fleur Delacour', 'Viktor Krum']	`LIST0=List(`
	`query=‘Triwizard Tournament champions’,`
	`max=4)`

[{'box': [213, 249, 299, 352], 'tag': 'Fleur Delacour', 'score': 100.0}, {'box': [98, 56, 182, 166], 'tag': 'Cedric Diggory', 'score': 99.5}, {'box': [476, 60, 560, 162], 'tag': 'Viktor Krum', 'score': 99.9}, {'box': [287, 142, 365, 233], 'tag': 'Harry Potter', 'score': 92.2}]	`OBJ1=Classify(`
	`image=IMAGE,`
	`categories=LIST0)`

	`IMAGE0=Tag(image=IMAGE, objects=OBJ1)`

Tag Brands

Tag these laptops by popular laptop companies in the US

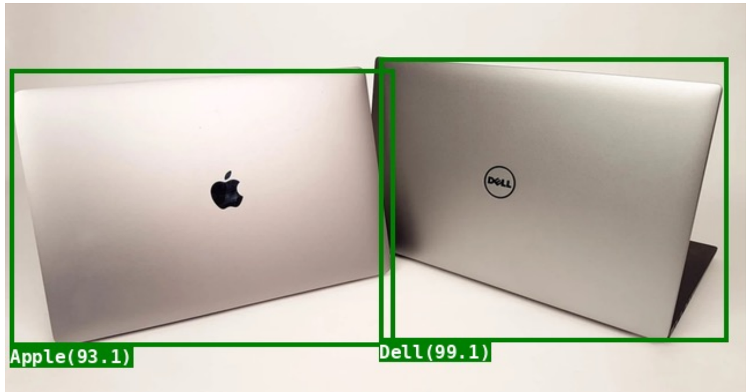


	IMAGE
	OBJ0=Loc(image=IMAGE, object=‘laptop’)

['Apple', 'Microsoft', 'Dell', 'Lenovo', 'Asus', 'Acer', 'HP', 'Samsung', 'Sony', 'Toshiba', 'LG', 'MSI', 'Razer', 'Alienware', 'Google', 'Huawei', 'Xiaomi']	LIST0=List(
	query=‘windows laptop manufacturers’,
	max=20)

[{'box': [4, 57, 336, 297], 'tag': 'Apple', 'score': 93.1}, {'box': [323, 47, 624, 293], 'tag': 'Dell', 'score': 99.1}]	OBJ1=Classify(
	image=IMAGE,
	categories=LIST0)

	IMAGE0=Tag(image=IMAGE, objects=OBJ1)

Tag Brands

Tag the car logos of these top
5 German car companies

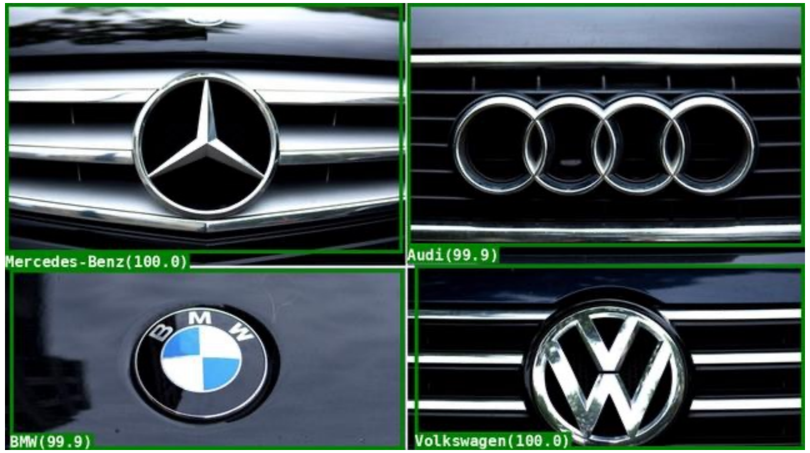


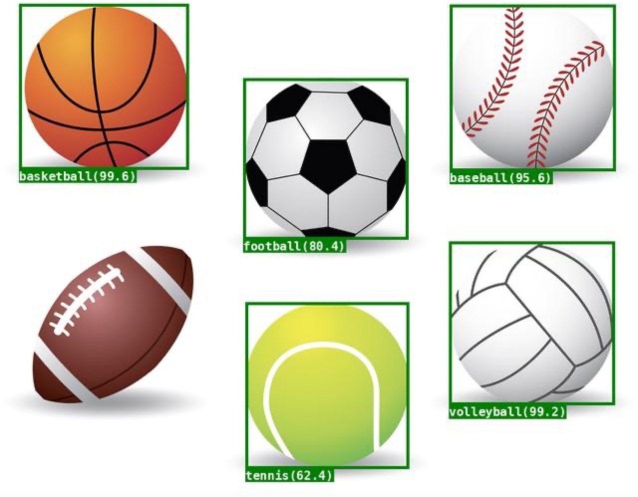
	IMAGE
	OBJ0=Loc(image=IMAGE, object=‘car logo’)

['BMW', 'Mercedes-Benz', 'Audi', 'Porsche', 'Volkswagen']	LIST0=List(
	query=‘top 5 German car companies’,
	max=5)

[{'box': [5, 279, 418, 468], 'tag': 'BMW', 'score': 99.9}, {'box': [0, 0, 417, 262], 'tag': 'Mercedes-Benz', 'score': 100.0}, {'box': [430, 274, 839, 468], 'tag': 'Volkswagen', 'score': 100.0}, {'box': [422, 0, 839, 255], 'tag': 'Audi', 'score': 99.9}]	OBJ1=Classify(image=IMAGE, categories=LIST0)
	IMAGE0=Tag(image=IMAGE, objects=OBJ1)

Tag Objects

Tag the spherical balls with popular ball-based sports



	`IMAGE`

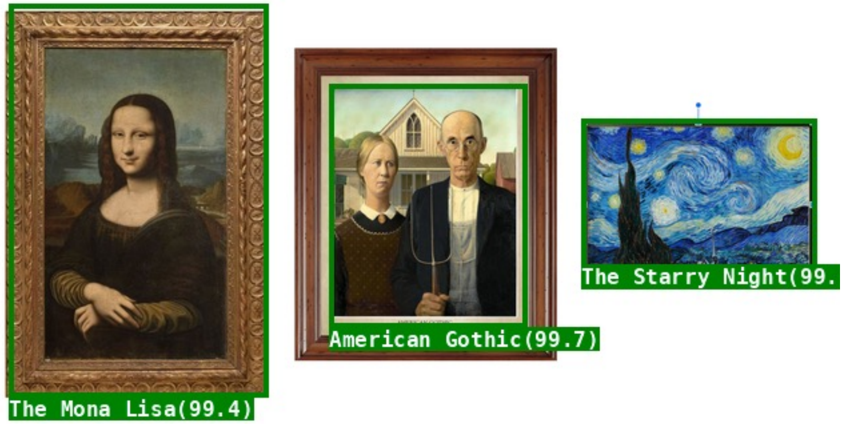
	`OBJ0=Loc(`
	`    image=IMAGE,`
	`    object=‘spherical ball’)`

`['baseball', 'basketball', 'football', 'golf', 'hockey', 'lacrosse', 'rugby', 'softball', 'volleyball', 'water polo', 'handball', 'jai alai', 'racquetball', 'squash', 'table tennis', 'tennis', 'ultimate frisbee', 'cricket', 'croquet']`	`LIST0=List(`
	`    query=‘popular ball-based sports’,`
	`    max=20)`

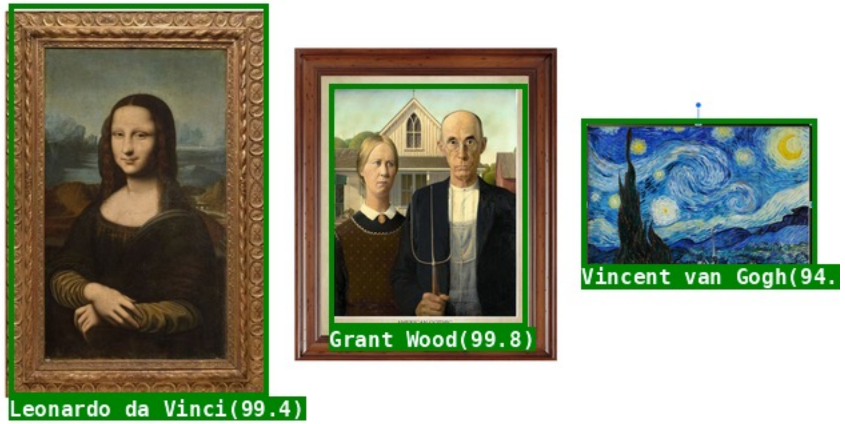
`[{'box': [25, 43, 248, 261], 'tag': 'basketball', 'score': 99.6}, {'box': [592, 44, 810, 263], 'tag': 'baseball', 'score': 95.6}, {'box': [323, 436, 538, 655], 'tag': 'tennis', 'score': 62.4}, {'box': [320, 141, 537, 353], 'tag': 'football', 'score': 80.4}, {'box': [591, 356, 809, 571], 'tag': 'volleyball', 'score': 99.2}]`	`OBJ1=Classify(image=IMAGE, categories=LIST0)`
	`IMAGE0=Tag(image=IMAGE, objects=OBJ1)`

Tag the same image in different ways

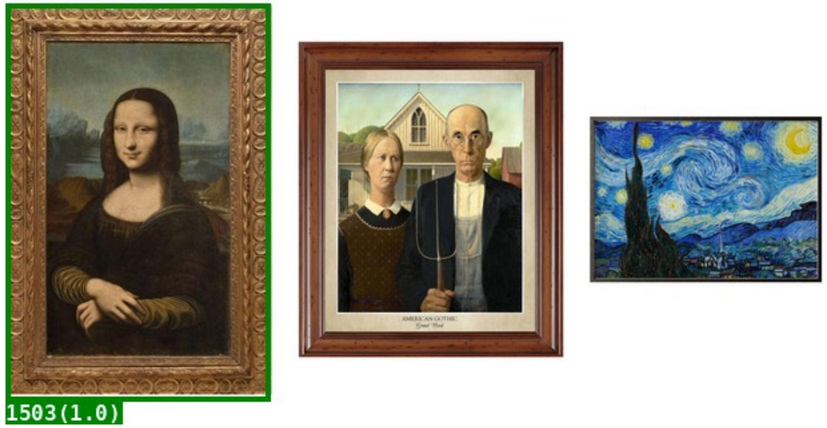
Tag these famous paintings



Tag these paintings with famous painters

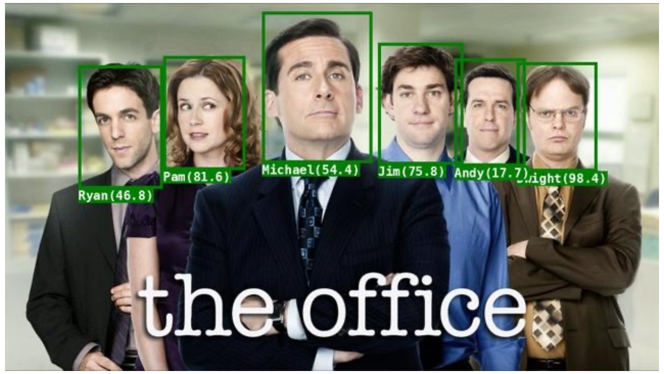


Tag ‘The Mona Lisa’ with the year Mona Lisa was painted

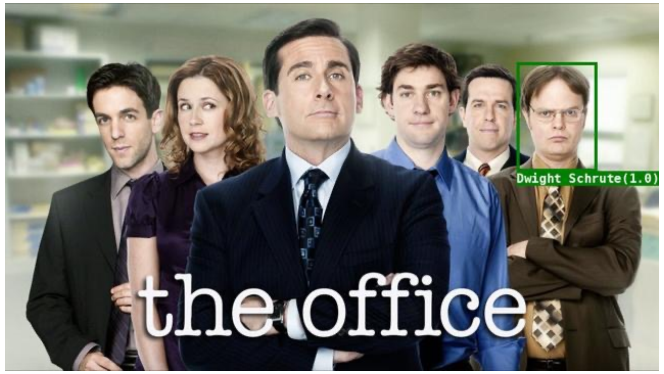


Tag the same image in different ways

Tag the characters from popular TV series The Office



Tag the owner of Schrute Farms from poular TV series The Office



Tag the wife of Jim Halpert in popular TV series The Office

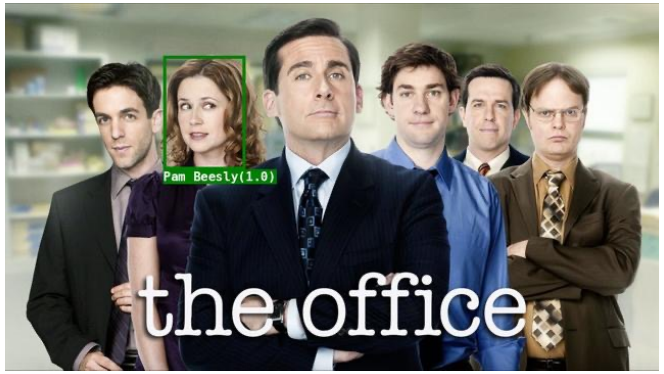


Image Editing with Natural Language

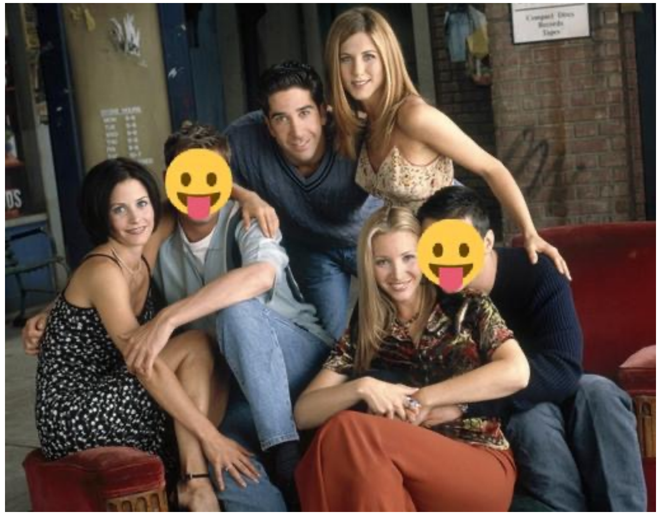
✓ Person De-identification

✓ Stylistic Object Highlighting

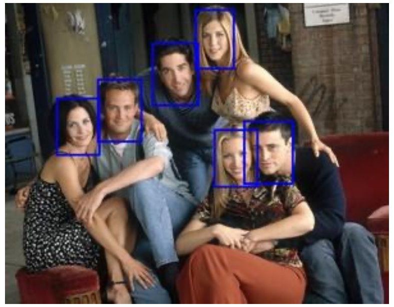
✓ Face & Object Replacement

Person De-identification

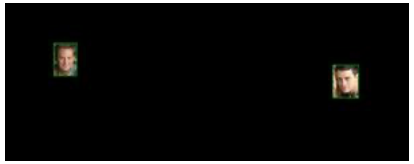
Hide Chandler and Joey with :ps



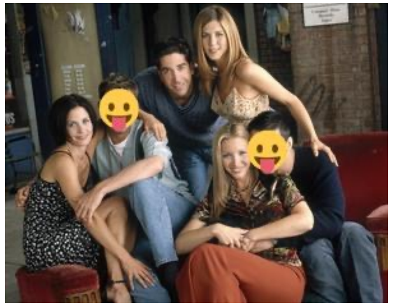
```
IMAGE
```



```
OBJ0=FaceDet(image=IMAGE)
```



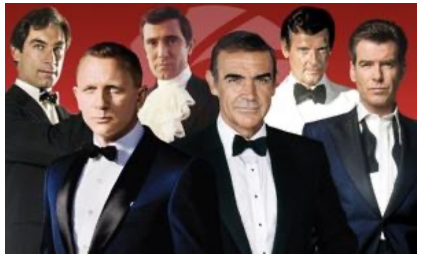
```
OBJ1=Select(
    image=IMAGE, object=OBJ0,
    query=['Chandler','Joey’],
    category=None)
```



```
IMAGE0=Emoji(
    image=IMAGE, object=OBJ1,
    emoji=‘face_with_tongue’)
```

Person De-identification

Hide Daniel Craig with 8), Sean Connery with ;), and Pierce Brosnan with :p



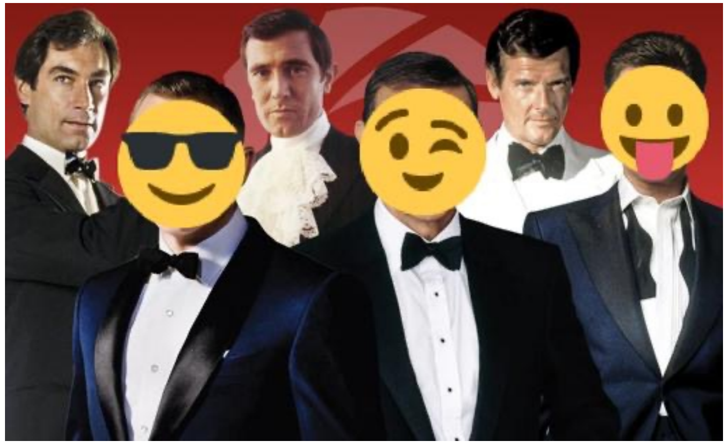
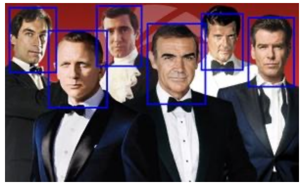
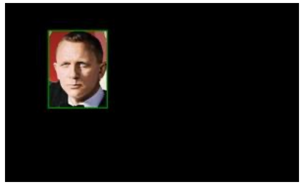
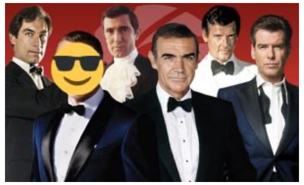
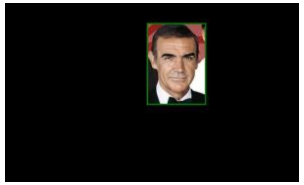
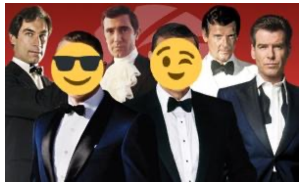


	IMAGE
	OBJ0=FaceDet(image=IMAGE)

	OBJ1=Select(
	image=IMAGE, object=OBJ0,
	query=[‘Daniel Craig’],
	category=None)

	IMAGE0=Emoji(
	image=IMAGE, object=OBJ1,
	emoji=‘smiling_face_with_sunglasses’)

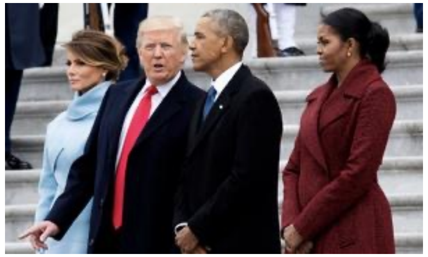
	OBJ2=Select(
	image=IMAGE, object=OBJ0,
	query=[‘Sean Connery’],
	category=None)

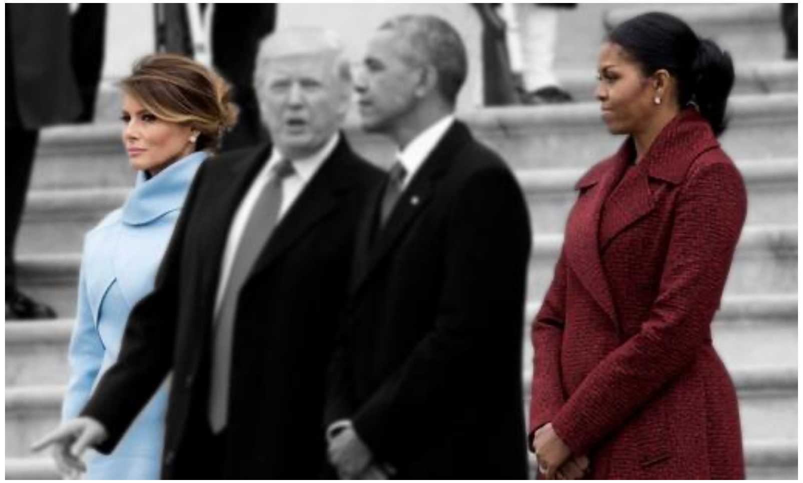
	IMAGE1=Emoji(
	image=IMAGE0, object=OBJ2,
	emoji=‘winking_face’)

	Steps for ‘Pierce Brosnan’ are similar to previous steps and not shown for brevity

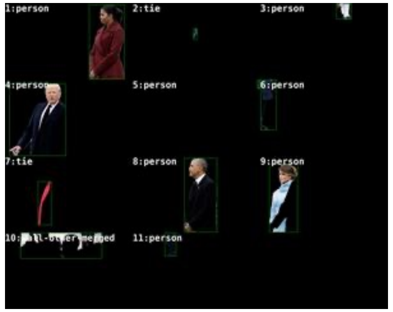
Stylistic Object Highlighting

Create a color pop of Melania Trump (person) and Michelle Obama (person) and blur the background

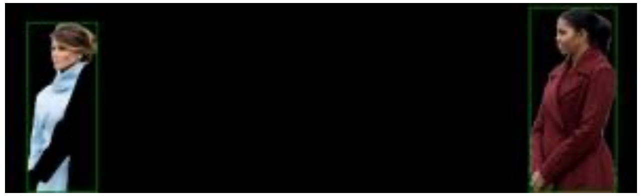




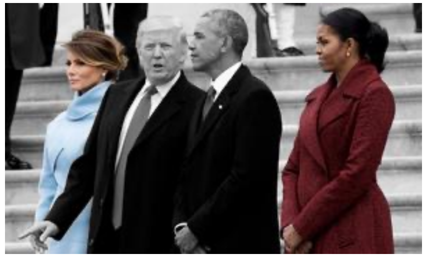
```
IMAGE
```



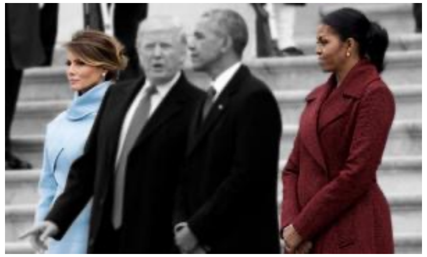
```
OBJ0=Seg(image=IMAGE)
```



```
OBJ1=Select(
    image=IMAGE, object=OBJ0,
    query=[‘Melania Trump’, ‘Michelle Obama’],
    category=‘person’)
```



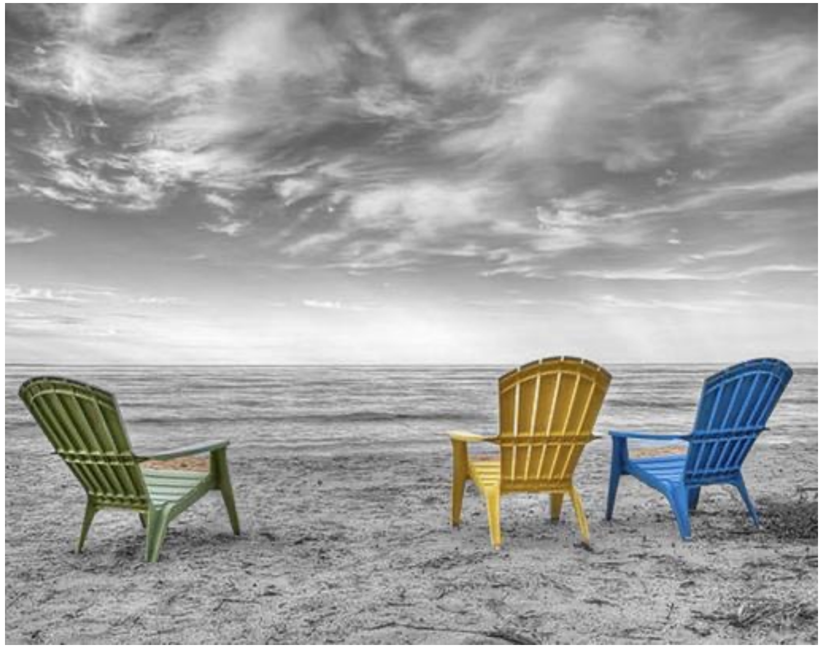
```
IMAGE0=ColorPop(image=IMAGE, object=OBJ1)
```

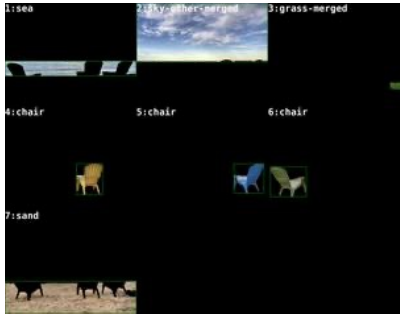


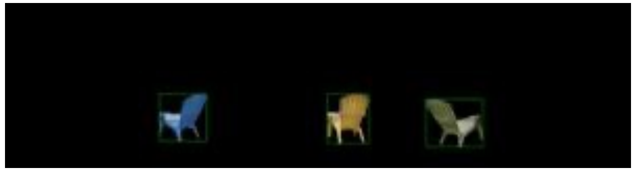
```
IMAGE1=BgBlur(image=IMAGE0, object=OBJ1)
```

Stylistic Object Highlighting

Create a color pop of blue, yellow, green chairs (chair)



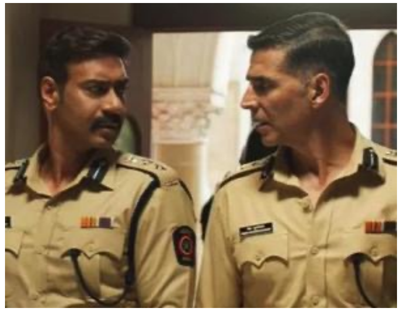
	`IMAGE`
 1:sea 2:sky-other-merged 3:grass-merged 4:chair 5:chair 6:chair 7:sand	`OBJ0=Seg(image=IMAGE)`

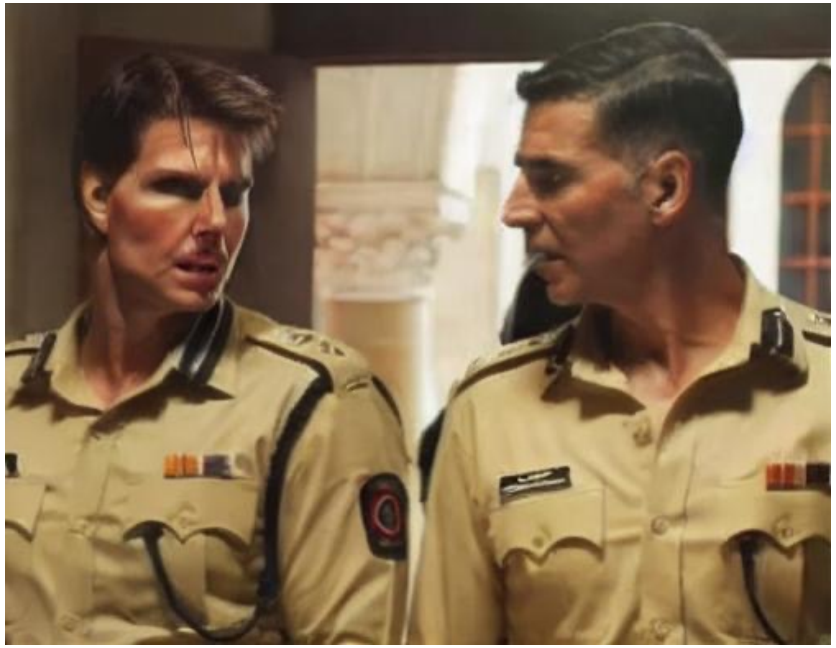
	`OBJ1=Select(`
	`image=IMAGE, object=OBJ0,`
	`query=[‘blue’, ‘yellow’, ‘green chairs’],`
	`category=‘chair’)`

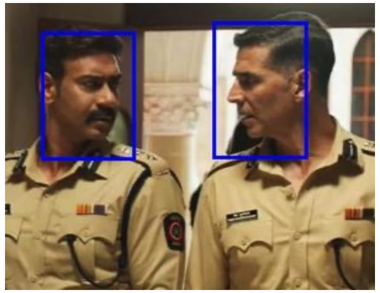
	`IMAGE0=ColorPop(image=IMAGE, object=OBJ1)`

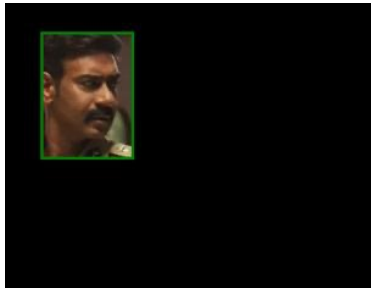
Person Replacement

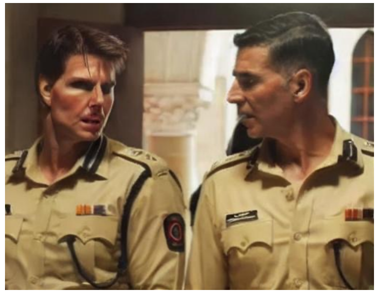
Replace Ajay Devgan with Tom Cruise





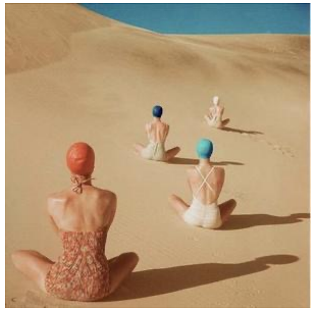
	`IMAGE`
	`OBJ0=FaceDet(image=IMAGE)`

	`OBJ1=Select(`
	`image=IMAGE, object=OBJ0,`
	`query=[‘Ajay Devgan’],`
	`category=None)`

	`IMAGE0=Replace(`
	`image=IMAGE,`
	`object=OBJ1,`
	`prompt=‘Tom Cruise’)`

Object Replacement

Replace the desert by sandy beach



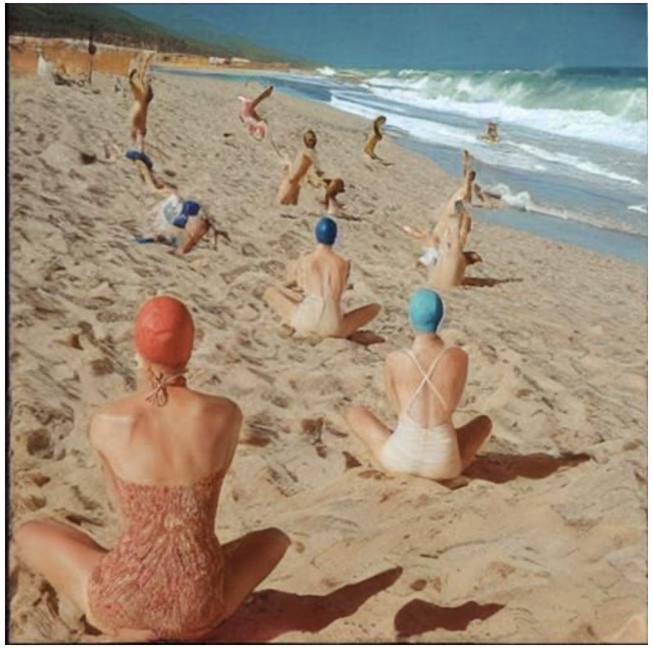
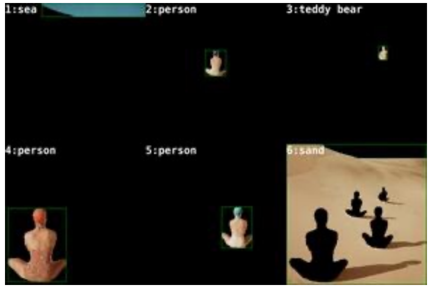
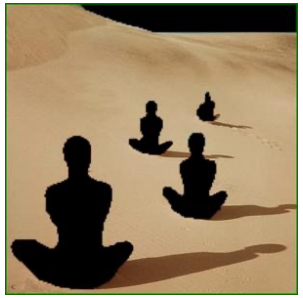
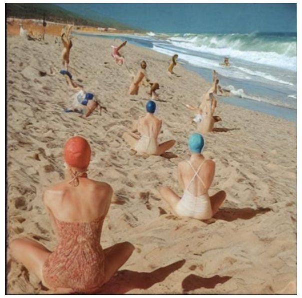


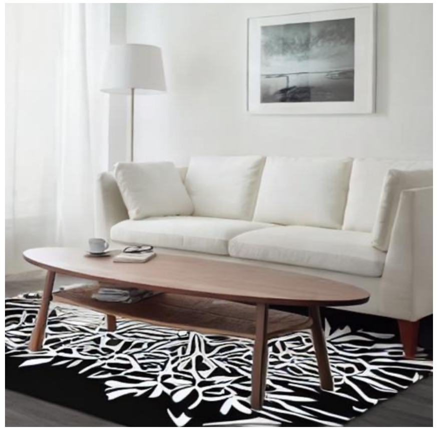
	IMAGE
 1:sea 2:person 3:teddy bear 4:person 5:person 6:sand	OBJ0=Seg(image=IMAGE)

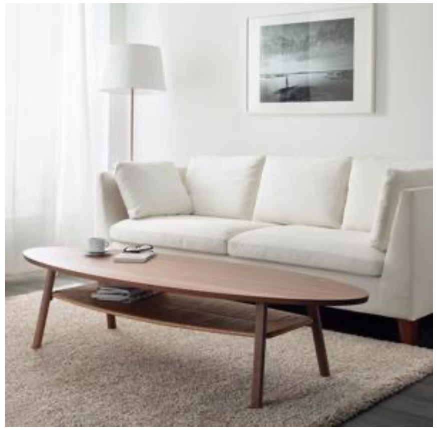
	OBJ1=Select(
	    image=IMAGE, object=OBJ0,
	    query=[‘desert’],
	    category=None)

	IMAGE0=Replace(
	    image=IMAGE,
	    object=OBJ1,
	    prompt=‘sandy beach’)

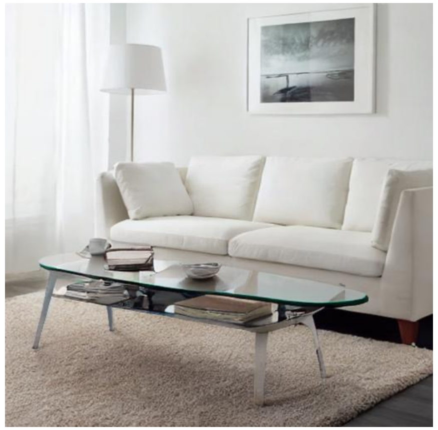
Edit the same image in different ways

Replace the rug with a black and white patterned carpet

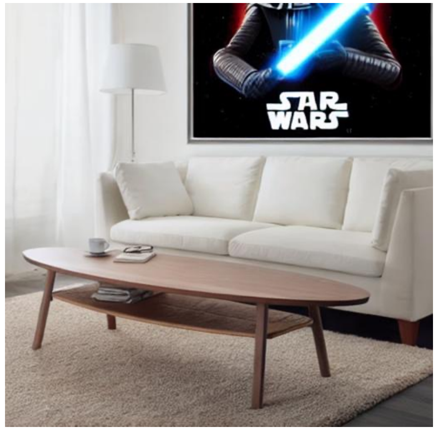




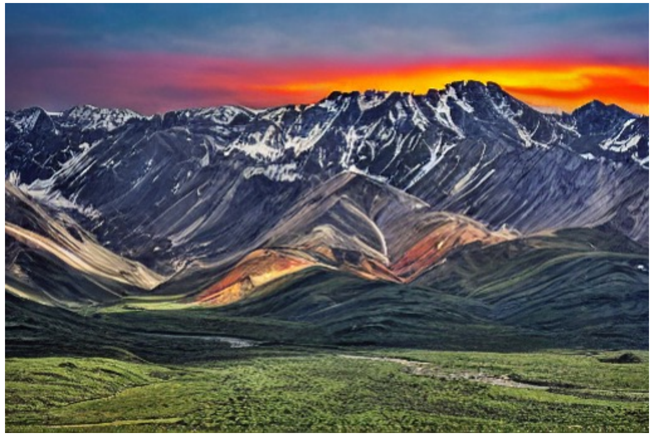
Replace the coffee table (table-merged) with a glass-top modern coffee table



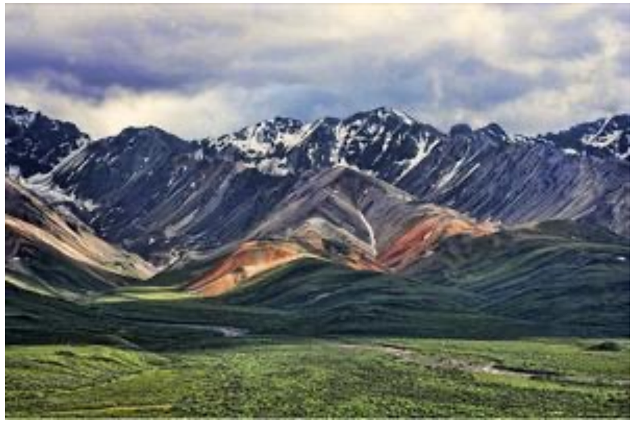
Replace the wall with a star wars poster hanging on the wall



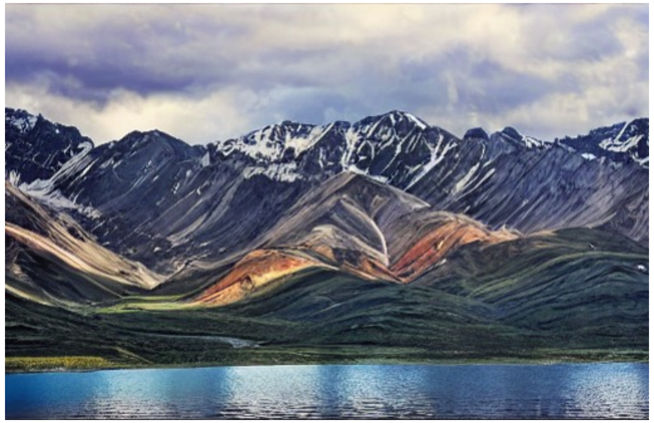
Edit the same image in different ways



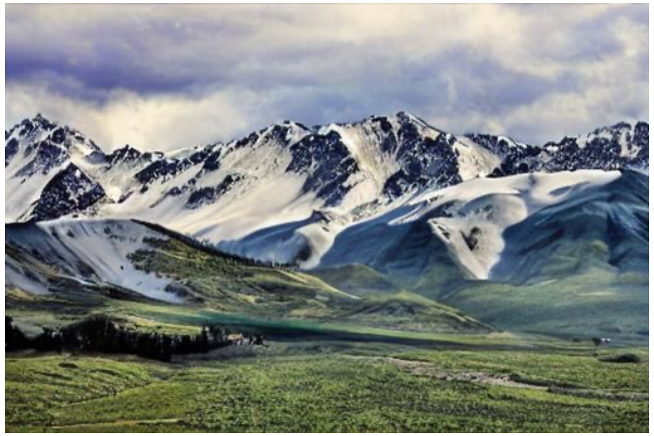
Replace the sky with sunrise behind mountains



Replace the grass with clear blue lake



Replace the rocky mountain (mountain-merged) with snowy mountain



Failure Cases

- Logically incorrect program
- Incorrect prediction from a module

Incorrect Programs

What is the piece of furniture that is made of same material as the shelf the book is on called?

```
BOX0=Loc(image=IMAGE, object=‘shelf’)
IMAGE0=Crop(image=IMAGE, bbox=BOX0)
BOX1=Loc(image=IMAGE, object=‘book’)
IMAGES=CropBelow(image=IMAGE0, bbox=BOX)
ANSWER0=Vqa(
    image=IMAGE1,
    question=‘What is the piece of furniture that is made
    of same material as the shelf the book is on called?’)
RESULT=ANSWER0

# Incorrect program – the correct program should first identify the
material of the shelf and then ask for the furniture that is made
of that material. The GQA prompts did not contain such examples
where a VQA question is created using the output of previous steps.
```

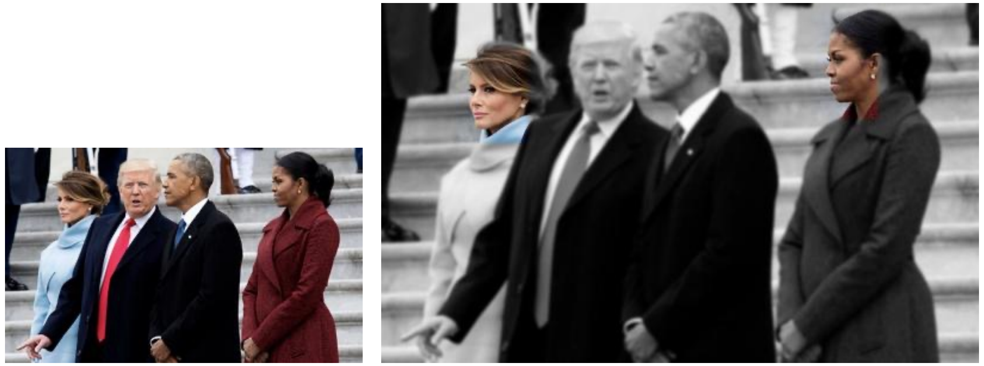
Is there a cow or a horse that is not white?

```
BOX0=Loc(image=IMAGE, object=‘cow’)
BOX1=Loc(image=IMAGE, object=‘horse’)
ANSWER0=Count(bbox=BOX0)
ANSWER1=Count(bbox=BOX1)
ANSWER1=Eval(“‘yes’ if {ANSWER0}+{ANSWER1}>0 else ‘no’”)
RESULT=ANSWER1

# The program does not check if the cow or horse is white
```

Incorrect Programs

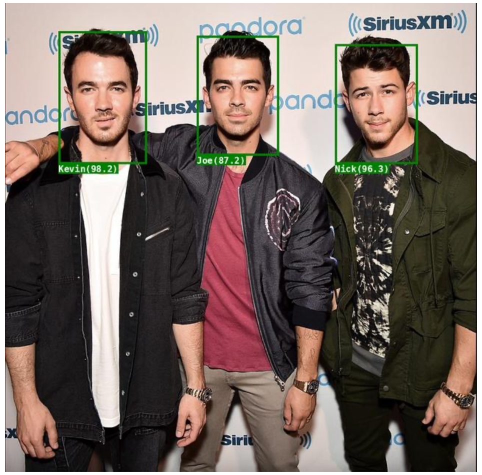
Create a color pop of Melania Trump and Michelle Obama and blur the background



```
OBJ0=FaceDet(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0,
            query=[‘Melania Trump’, ‘Michelle Obama’],
            category=None)
IMAGE0=ColorPop(image=IMAGE, object=OBJ1)
IMAGE1=BgBlur(image=IMAGE0, object=OBJ1)
RESULT=IMAGE1

# Generated program uses FaceDet instead of Seg,
resulting in a color pop of only the faces instead of
whole body
```

Tag the Jonas brother who married the Indian actress Priyanka Chopra

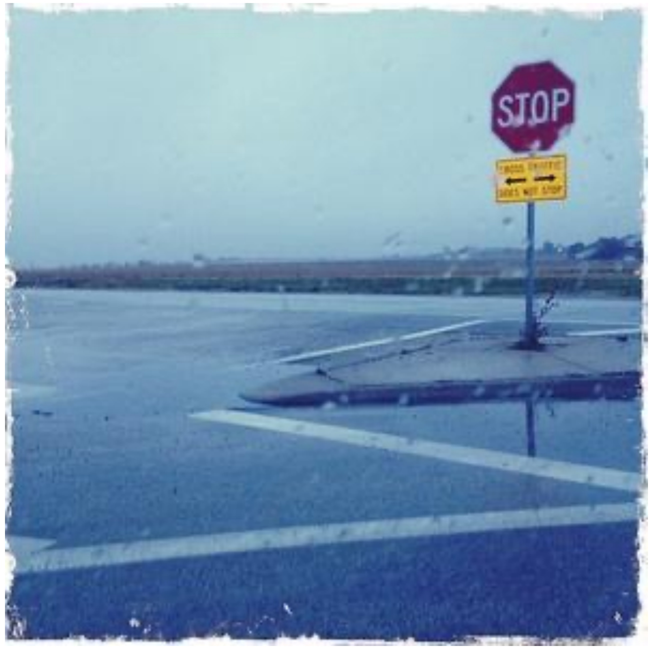


```
OBJ0=FaceDet(image=IMAGE)
LIST0=List(query=‘Jonas Brothers’, max=3)
OBJ1=Classify(image=IMAGE, object=LIST0)
IMAGE0=Tag(image=IMAGE, object=OBJ1)
RESULT=IMAGE0

# The query to List should be ‘Jonas Brother
who married the Indian actress Priyanka
Chopra’ and max=1
```

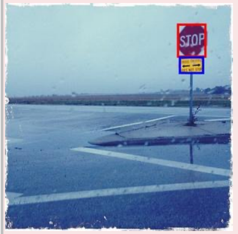
Module Failure

Is there a green stop sign or traffic light?



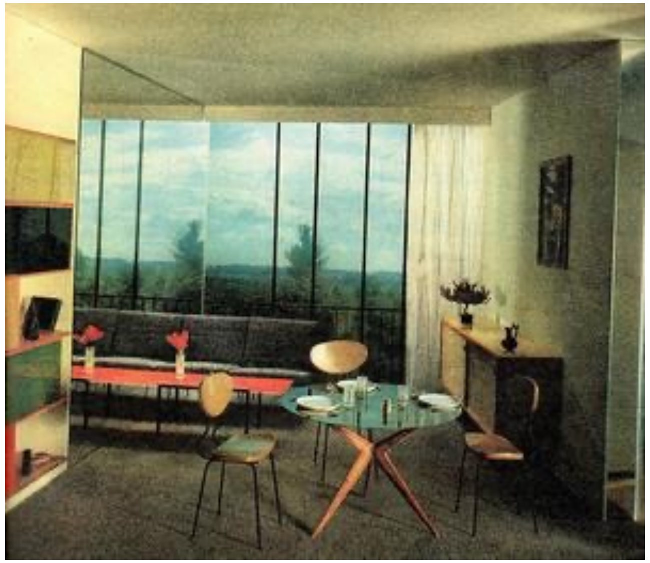
Prediction: yes

Label: no

	IMAGE
	BOX0=Loc(image=IMAGE, object=‘green stop sign’) # 2 boxes where predicted while there should be none
2	ANSWER0=Count(bbox=BOX0)
	BOX1=Loc(image=IMAGE, object=‘green traffic light’)
0	ANSWER1=Count(bbox=BOX1)
yes	ANSWER2=Eval(expr="'yes' if {ANSWER0}>0 or {ANSWER1}>0 else 'no’”) =Eval(expr="'yes' if 2>0 and 0>0 else 'no'")

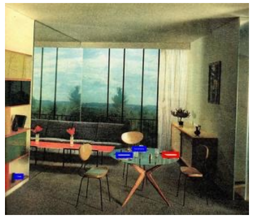
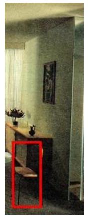
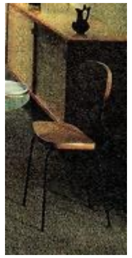
Module Failure

Is the chair to the right of the plate made out of wicker?



Prediction: yes

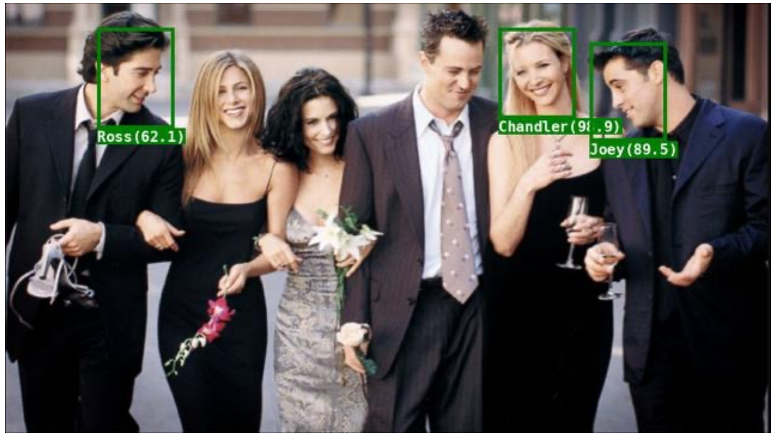
Label: no

	`IMAGE`
	`BOX0=Loc(image=IMAGE, object=‘plate’)`
	`IMAGE0=CropRight(image=IMAGE, bbox=BOX0)`
	`BOX1=Loc(image=IMAGE0, object=‘chair’)`
	`IMAGE1=Crop(image=IMAGE0, bbox=BOX1)`

yes	`ANSWER0=Vqa(image=IMAGE1,`
	`question=‘Is the chair made out of wicker?’)`

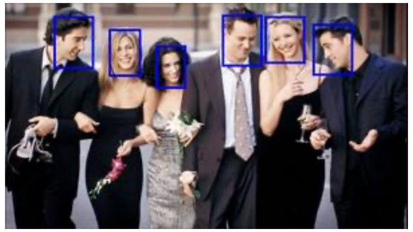
Module Failure

Tag the three male lead characters from Friends series.



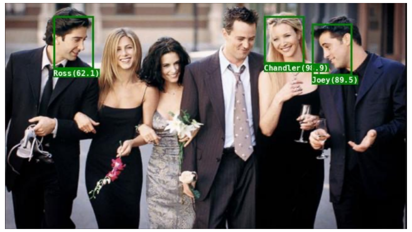
Prediction: yes

Label: no

	`IMAGE`
	`OBJ0=FaceDet(image=IMAGE)`

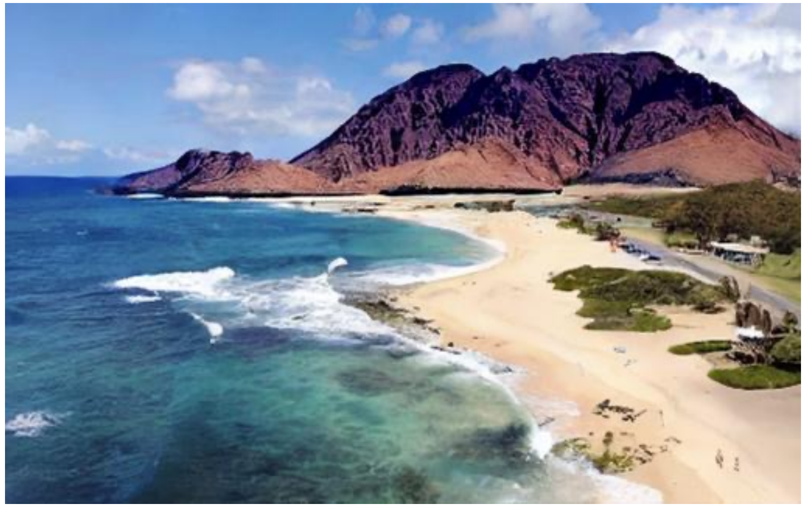
`[‘Chandler', ‘Joey', ‘Ross’]`	`LIST0=List(`
	`    query=‘male lead characters from Friends series’,`
	`    max=3)`

`[{'box': [101, 27, 185, 137], 'tag': 'Ross', 'score': 62.1}, {'box': [541, 27, 625, 126], 'tag': 'Chandler', 'score': 98.9}, {'box': [641, 43, 725, 151], 'tag': 'Joey', 'score': 89.5}]`	`OBJ1=Classify(image=IMAGE, categories=LIST0)`
	`# Incorrectly classified Phoebe as Chandler`

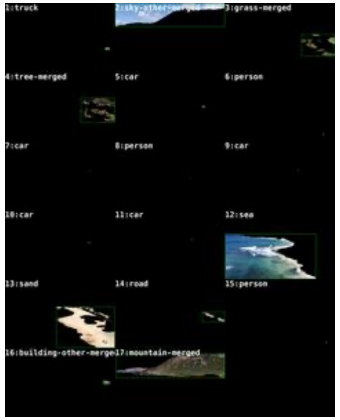
	`IMAGE0=Tag(image=IMAGE, objects=OBJ1)`

Module Failure

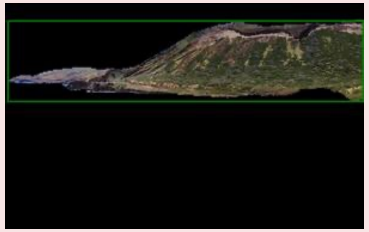
Replace the ocean by a sandy desert



```
IMAGE
```

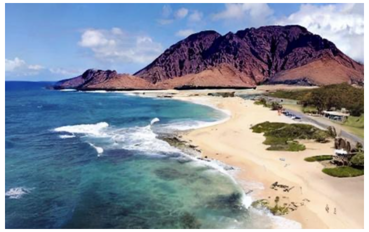


```
OBJ0=Seg(image=IMAGE)
```



```
OBJ1=Select(
    image=IMAGE, object=OBJ0,
    query=['ocean'],
    category=None)

# selected mountain instead of ocean
```



```
IMAGE0=Replace(
    image=IMAGE,
    object=OBJ1,
    prompt='sandy desert')
```

Thank You!