```
In [1]:  # Set up matplotlib.
         %matplotlib inline
```
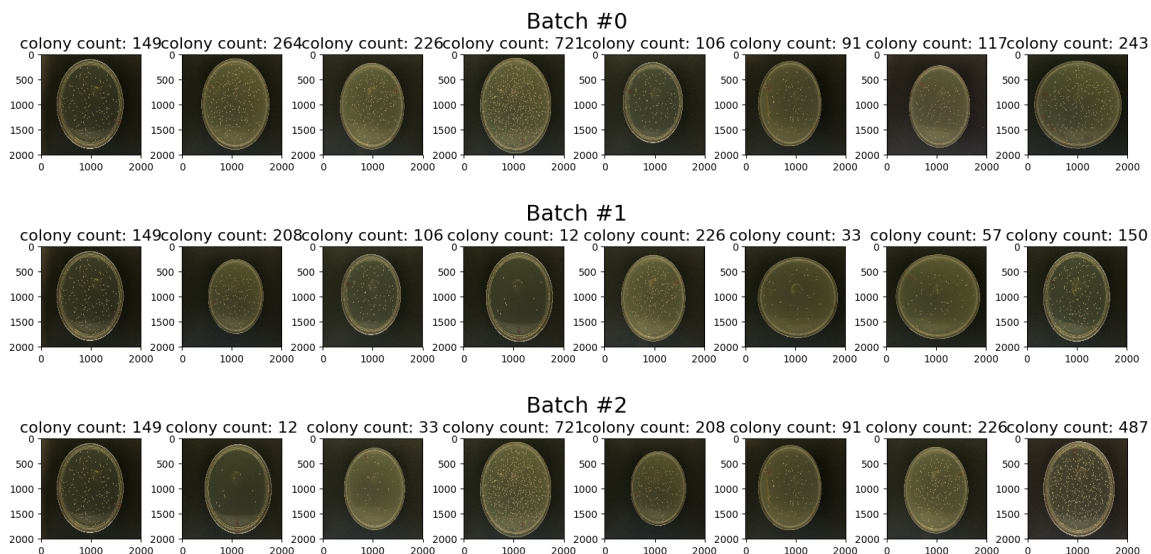
```
In [30]:  # Import our package.
          import sys, importlib
          sys.path.append("/home/ubuntu/cell_counting")

          from src import dataset, visualization
          importlib.reload(dataset)
          importlib.reload(visualization)
```

```
Out[30]:  <module 'src.visualization' from '/home/ubuntu/cell_counting/src/visualization.p
          y'>
```

```
In [3]:  # Load the dataset, processing it as a collection of image-count pairs.
         images_counts = dataset.Dataset(3)
         images_counts.load_images_and_excel_labels("/home/ubuntu/cell_counting/data/easy/r
         aw/images",
                                                    "/home/ubuntu/cell_counting/data/easy/r
         aw/Plates.xlsx", "A", "C",
                                                    (2000, 2000))
```

```
In [4]:  # Plot a few batches.
         for batch in range(3):
             inputs, outputs = images_counts.get_batch(8)
             visualization.show_image_grid(inputs, 1, 8, 2.5, 16, "Batch #{0}".format(batch
         ),
                                          ["colony count: {0}".format(count) for count in
         outputs])
```

Batch #0

colony count: 149 colony count: 264 colony count: 226 colony count: 721 colony count: 106 colony count: 91 colony count: 117 colony count: 243

Batch #1

colony count: 149 colony count: 208 colony count: 106 colony count: 12 colony count: 226 colony count: 33 colony count: 57 colony count: 150

Batch #2

colony count: 149 colony count: 12 colony count: 33 colony count: 721 colony count: 208 colony count: 91 colony count: 226 colony count: 487

In [31]:
```
# Load the dataset, processing it as a collection of image-mask pairs.
images_masks = dataset.Dataset(1)
images_masks.load_image_mask_pairs("/home/ubuntu/cell_counting/data/easy/raw/image
s",
                                   "/home/ubuntu/cell_counting/data/easy/raw/masks
", (2000, 2000))
```

In [32]:
```
# Plot a batch.
inputs, outputs = images_masks.get_batch(3)
visualization.show_image_grid(inputs, 1, 3, 2, 6, "images")
visualization.show_image_grid(outputs, 1, 3, 2, 6, "masks")
```
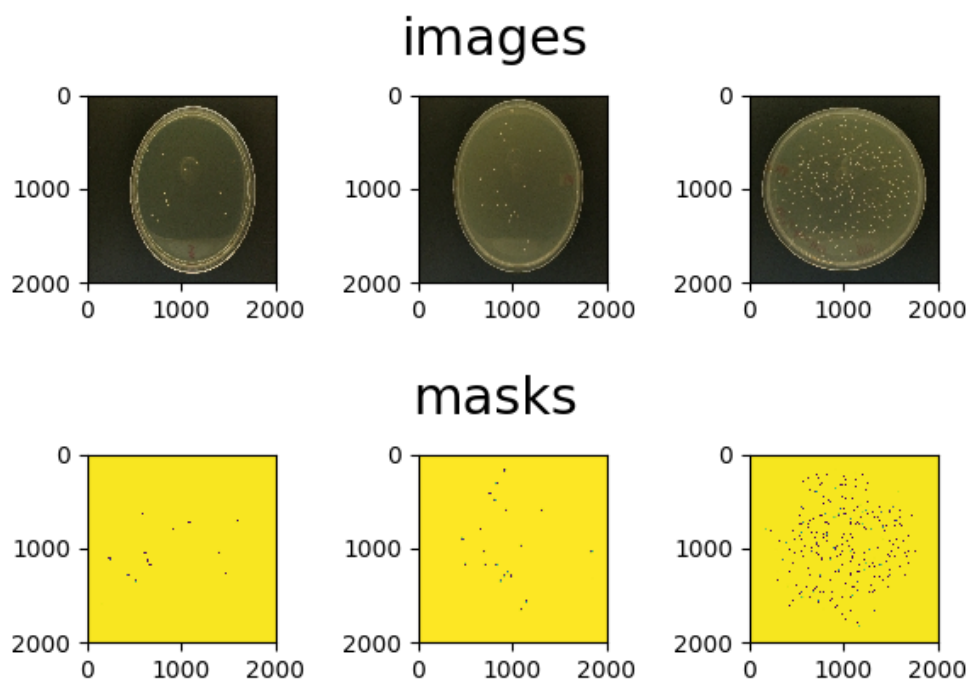

In [ ]:
```
# Close the datasets.
images_counts.close()
images_masks.close()
```