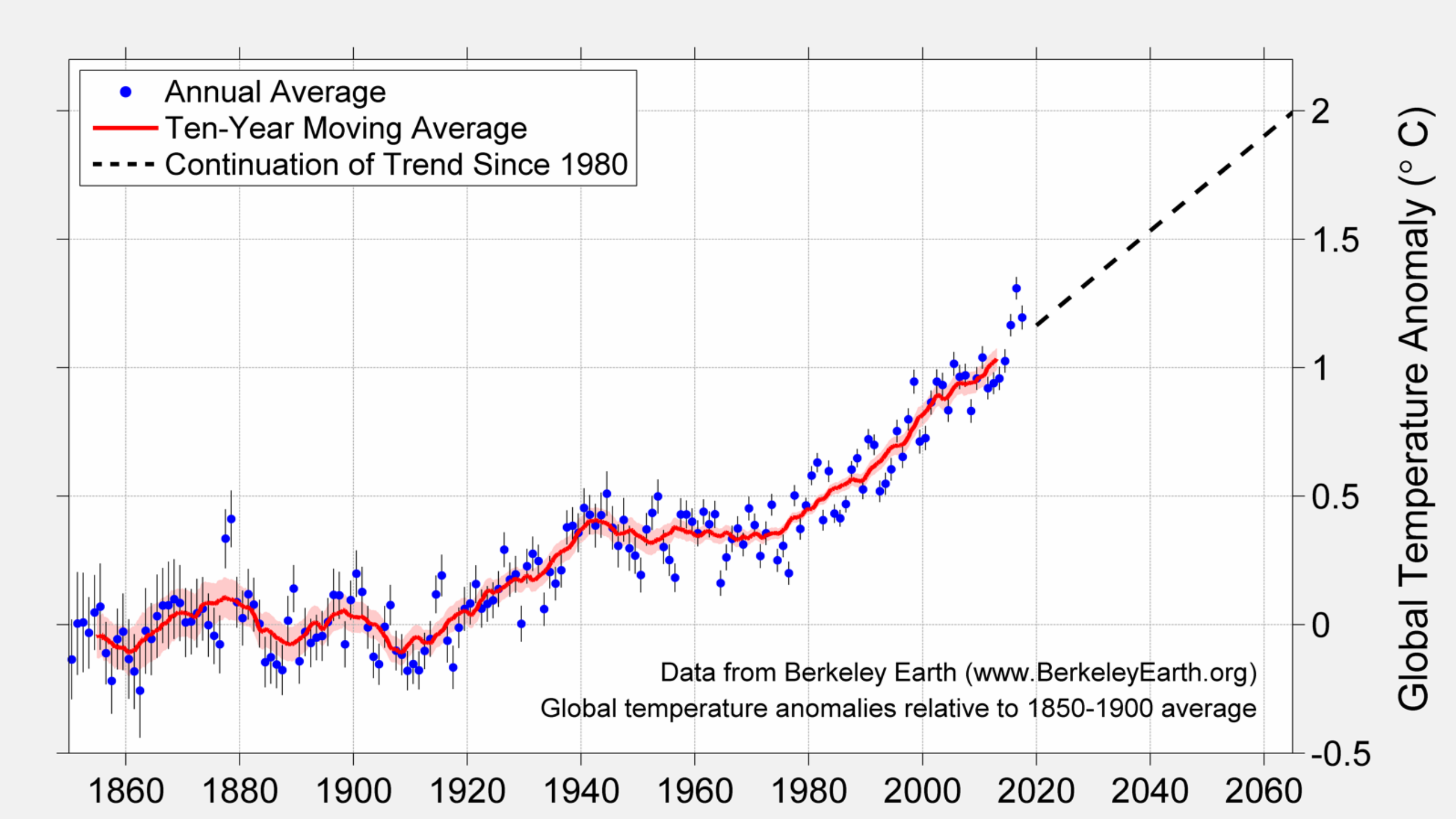
온난화팀

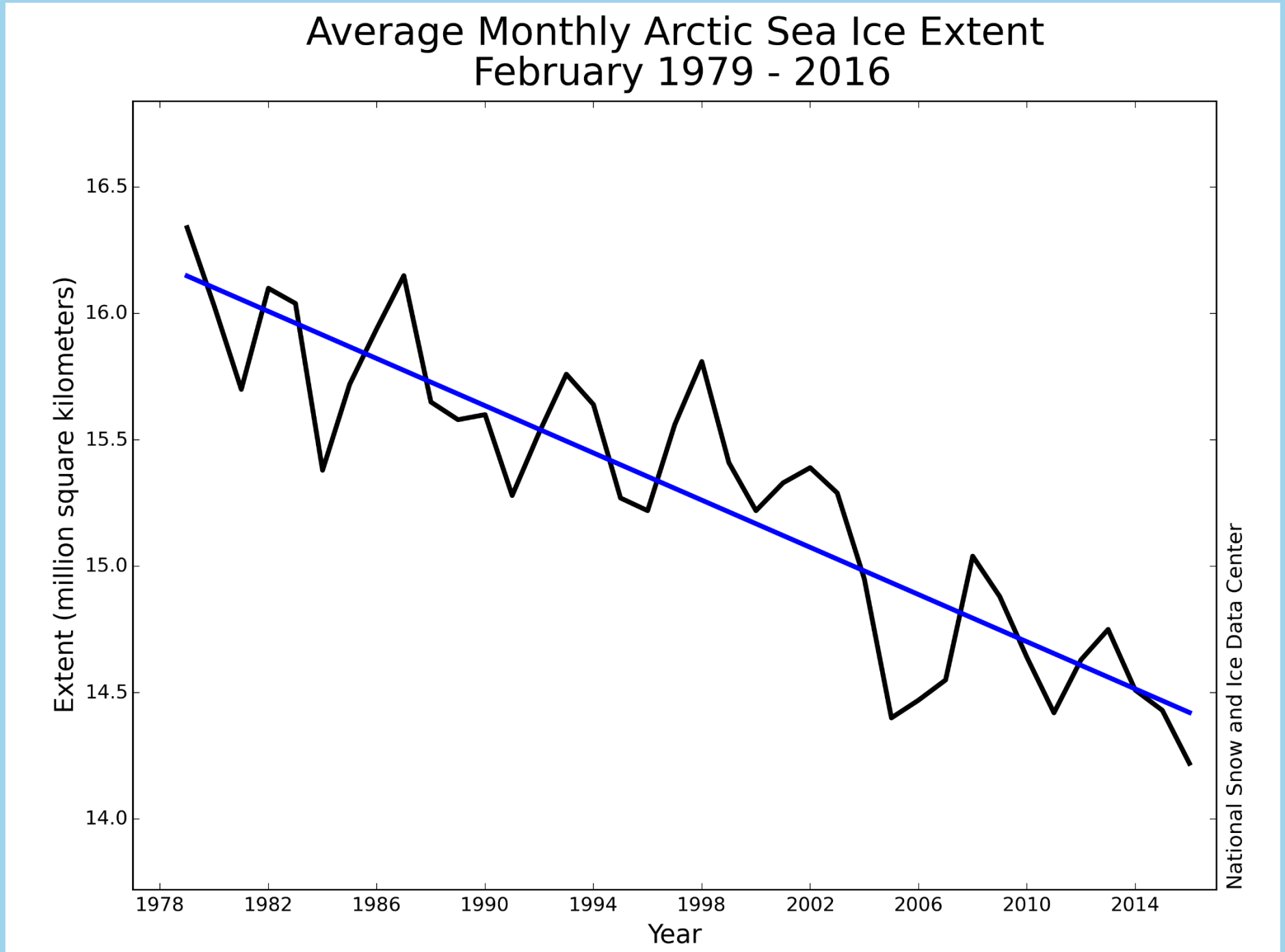
권다연 백서연 윤종필 이강산


목차
문제제기
크롤링
모델소개
ICONEO

Polar bear cubs drowning due to sea ice loss, says report

Polar bear cubs in Alaska are drowning on long swims with their mothers because of melting sea ice, a new study finds


Annual Average
Ten-Year Moving Average
Continuation of Trend Since 1980
Data from Berkeley Earth (www.BerkeleyEarth.org)
Global temperature anomalies relative to 1850-1900 average
Global Temperature Anomaly (° C)
2
1.5
1
0.5
0
-0.5
1860
1880
1900
1920
1940
1960
1980
2000
2020
2040
2060


Average Monthly Arctic Sea Ice Extent
February 1979 - 2016
Extent (million square kilometers)
16.5
16.0
15.5
15.0
14.5
14.0
1978
1982
1986
1990
1994
1998
2002
2006
2010
2014
Year
National Snow and Ice Data Center


Kerstin Langenberger
www.arctic-dreams.com

크롤링

```
#!pip install google_images_download

from google_images_download import google_images_download

import ssl
ssl._create_default_https_context = ssl._create_unverified_context


def imageCrawling(keyword, dir):

  response = google_images_download.googleimagesdownload()

  arguments = {"keywords":"winter fashion",
               "limit":101,
               "print_urls":True,
               "no_directory" : True,
               "output_directory" : dir}
  paths = response.download(arguments)
  print(paths)
```

```
links = []
for i in range(0, 100):

    link = (fnd[i].get('src', ''))
    if link == '':
        print('its empty')
    else:
        links.append(link)

# for i in range(0, 100):

#     link = (fnd[i].get('data-src', ''))
#     if link == '':
#         print('its empty')
```

Glacier / Polar Bear 학습

```
X = tf.placeholder(tf.float32, [None, 64, 64, 3])
Y = tf.placeholder(tf.float32, [None, 2])
```

```
W1 = tf.Variable(tf.random_normal([3,3,3,32], stddev=0.01))
L1 = tf.nn.conv2d(X, W1, strides=[1,1,1,1], padding='SAME')
L1 = tf.nn.relu(L1)
L1 = tf.nn.max_pool(L1, ksize=[1,2,2,1], strides=[1,2,2,1], padding='SAME')
print(L1)
```

```
Tensor("MaxPool_3:0", shape=(?, 32, 32, 32), dtype=float32)
```

```
W2 = tf.Variable(tf.random_normal([3,3,32,64], stddev=0.01))
L2 = tf.nn.conv2d(L1, W2, strides=[1,1,1,1], padding='SAME')
L2 = tf.nn.relu(L2)
L2 = tf.nn.max_pool(L2, ksize=[1,2,2,1], strides=[1,2,2,1], padding='SAME')
print(L2)
```

```
Tensor("MaxPool_4:0", shape=(?, 16, 16, 64), dtype=float32)
```

```
W3 = tf.Variable(tf.random_normal([3,3,64,128], stddev=0.01))
L3 = tf.nn.conv2d(L2, W3, strides=[1,1,1,1], padding='SAME')
L3 = tf.nn.relu(L3)
L3 = tf.nn.max_pool(L3, ksize=[1,2,2,1], strides=[1,2,2,1], padding='SAME')
print(L3)
```

```
Tensor("MaxPool_5:0", shape=(?, 8, 8, 128), dtype=float32)
```

```
W4 = tf.Variable(tf.random_normal([8*8*128, 256], stddev=0.01))
L4 = tf.reshape(L3, [-1,8*8*128])
L4 = tf.matmul(L4, W4)
L4 = tf.nn.relu(L4)
print(L4)
```

```
Tensor("Relu_7:0", shape=(?, 256), dtype=float32)
```

```
W5 = tf.Variable(tf.random_normal([256,2], stddev=0.01))
model=tf.matmul(L4, W5)
model
```

```
<tf.Tensor 'MatMul_3:0' shape=(?, 2) dtype=float32>
```

```
cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(logits=model, labels=Y))
optimizer = tf.train.AdamOptimizer(0.0001).minimize(cost)
```

```
is_correct = tf.equal(tf.argma
accuracy = tf.reduce_mean(tf.c

print('train  : ', sess.run(ac
print('test   : ', sess.run(ac
```

```
train  :  0.96917146
test   :  0.76685935
```


glacier
bear
glacier
glacier
glacier
glacier
glacier
bear
bear
bear
glacier
bear
glacier
bear
glacier
bear
glacier
bear
bear
bear
glacier
glacier
bear
glacier
bear
bear
glacier