

1.

	COVID19-DL	LIDC-IDRI
link	https://github.com/adnan-saood/COVID19-DL	https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI
typ	CT	CT
format	png	dicom
liczba bitów	8	16
przestrzeń barwna	greyscale	greyscale
wielkość	14 147,42 B	351mm x 351mm x 1.9mm
rozdzielczość	256x256	512x512

2.

LIDC-IDRI

31% to zdjęcia z kontrastem

W artykule o tym zbiorze autorzy twierdzą, że wiek i płeć jest w metadanych, ale my ich nie znaleźliśmy.

COVID19-DL

brak jakichkolwiek informacji

3.

Another challenging characteristic of the databases is the uneven distribution of the considered classes across the cases. Table I provides statistics for the entire dataset, while figure 2 presents a sample CT lung slice along with the given annotations.

TABLE I: Data statistics across the considered classes i.e. Healthy (H), Ground Glass Opacity (GGO), Micronodules (MN), Consolidation (Cons), Reticulation (Ret) and Honey-combing (HC).

	H	GGO	MN	Cons	Ret	HC	Totals
#Pixels $\times 10^5$	92.5	27.7	35.8	7.08	28.2	20.1	211.4
#Cases	66	82	15	46	81	47	172

Secondly, the distribution of the considered classes in the dataset was highly imbalanced, a fact that can be challenging for any classification method. We tackled both problems by scaling the considered loss and accuracy with appropriate weighting schemes computed for each set. All pixels corresponding to annotated areas were assigned a weight inversely proportional to the number of samples of its class in the specific set. In this way, all classes contributed equally to the considered metrics

```
class wcceOA(object):
    ...
    Weighted categorical cross entropy over annotation mask
    ...
    def __init__(self, weights):
        self.config = {'weights': list(weights)}
        self.length = len(weights)
        self.weights = K.variable(weights, name='loss_weights')
        self.__name__ = 'wcceOA'

    def __call__(self, y_true, y_pred):
        weights_mask = K.zeros_like(y_true[:,:,:,:0])
        for c in range(self.length):
            weights_mask += self.weights[c] *
K.cast(y_true[:,:,:,:c]>0,K.floatx())
        self.weights_mask = weights_mask

        wcce = K.sum(-K.sum(y_true * K.log(y_pred),
axis=-1)*self.weights_mask) / K.sum(self.weights_mask+1e-8)

        return wcce

    def get_config(self):
        config = self.config
        return config
```

```

class wcce0A(object):
    """
    Weighted categorical cross entropy over annotation mask
    """
    def __init__(self, weights):
        self.config = {'weights': list(weights)}
        self.length = len(weights)
        self.weights = K.variable(weights, name='loss_weights')
        self.__name__ = 'wcce0A'

    def __call__(self, y_true, y_pred):
        weights_mask = K.zeros_like(y_true[:, :, :, 0])
        for c in range(self.length):
            weights_mask += self.weights[c] * K.cast(y_true[:, :, :, c] > 0, K.floatx())
        self.weights_mask = weights_mask

        wcce = K.sum(-K.sum(y_true * K.log(y_pred), axis=-1) * self.weights_mask) / K.sum(self.weights_mask + 1e-8)

        return wcce

    def get_config(self):
        config = self.config
        return config

```

4.

Sieć bierze dla każdej klasy (i tła) maskę w postaci logicznej macierzy dwuwymiarowej o wymiarach obrazkach.

Zbiór COVID19-DL zawierał maski w postaci obrazka, którego różne wartości odnosiły się do różnych klas. Porozdzielaliśmy je po unikalnych wartościach.

Zbiór LIDC-IDRI zawierał skany składające się z dziesiątek warstw (oś Z). Te warstwy traktowaliśmy niezależnie i wyfiltrowaliśmy tylko te, na których znajduje się jakaś adnotacja. W orginale maski były binarne i miały wymiar 512x512x<ilość warstw>, więc dla każdej warstwy wyciągneliśmy odpowiedni fragment 512x512 oraz dodaliśmy jego negację jako maskę tła.