UMSL | Department of
Mathematics &
Computer Science

27TH CONFERENCE ON
Intelligent Systems
for Molecular Biology
AND THE 18TH
European Conference
on Computational Biology

BASEL JULY 21-25 2019
ISMB ECCB
ISCB
3DSIG

DEEPCON: Protein Contact Prediction using Dilated Convolutional Neural Networks with Dropout

Badri Adhikari
adhikarib @ umsl.edu

Assistant Professor of CS
Department of Mathematics & Computer Science
University of Missouri-St. Louis

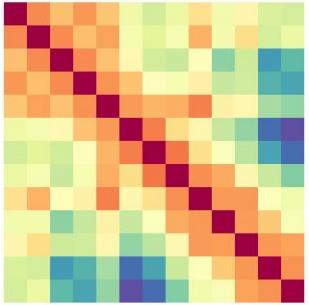
What is Contact Prediction?

A Protein’s Amino Acid Sequence

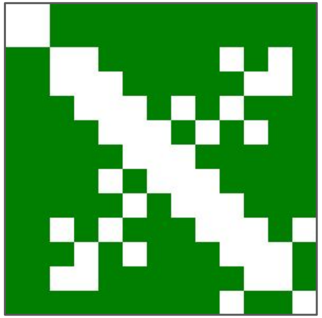
G F G C N G P W . . . D E D D M

?

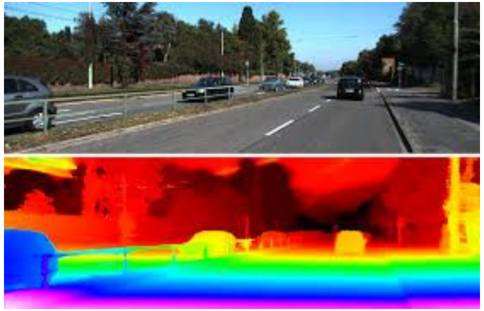
Prediction



Distance Map

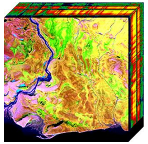


Contact Map

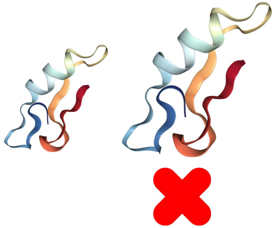


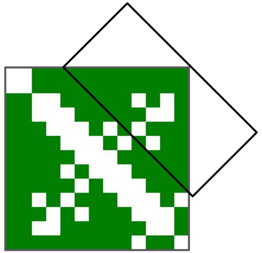
Depth Estimation / Image Segmentation

Features of the Contact Prediction Problem (ML perspective)



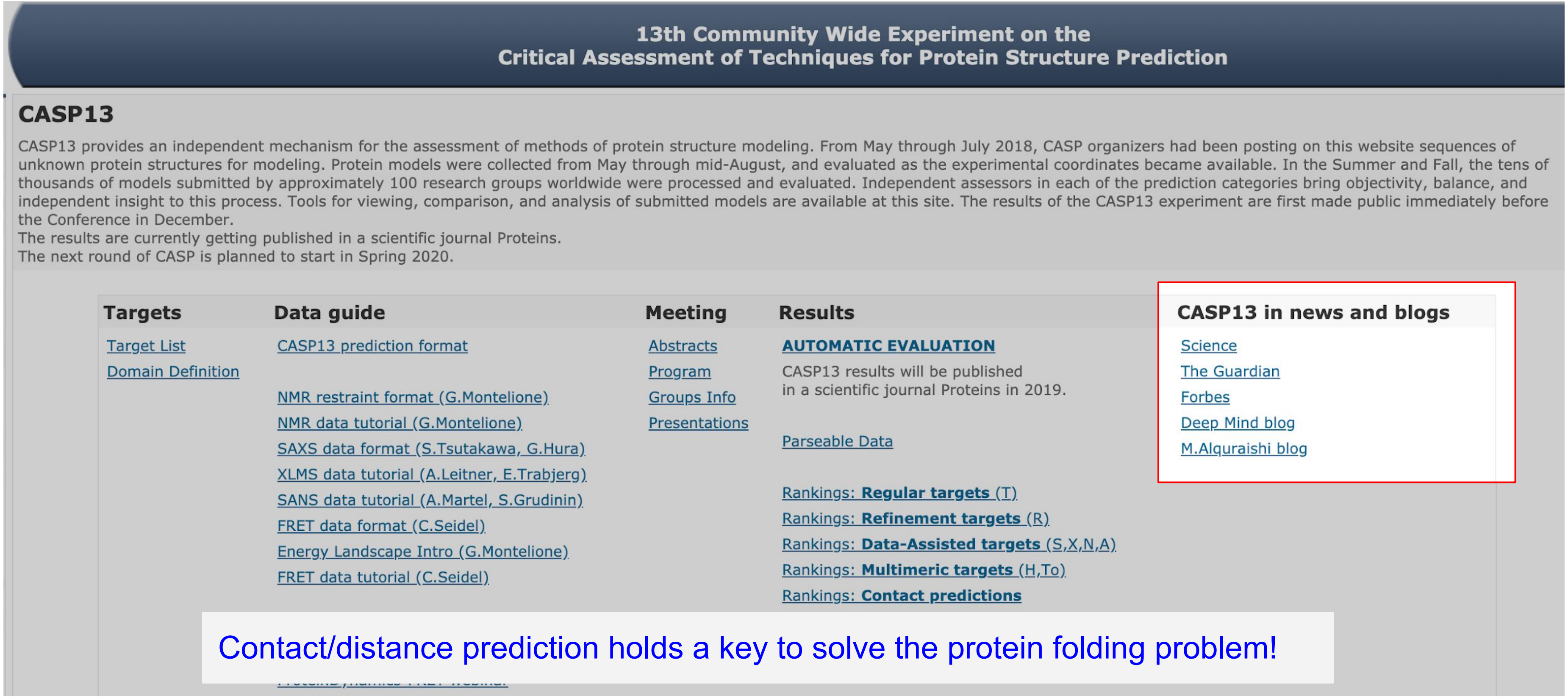
- Large number of input channels (like hyperspectral images)
- Input features are 0D, 1D, or 2D
- Visualization
- Non-scalability of protein 3D structures
- Variable feature size (L can be 1000+)
- Limited and unbalanced data (Yet, not all are needed)
- The goal is to predict distances (continuous distance values)
- Long-range contacts are important
- Symmetrical along diagonal





Why Predict Contacts?

http://predictioncenter.org/casp13/index.cgi



13th Community Wide Experiment on the
Critical Assessment of Techniques for Protein Structure Prediction

CASP13

CASP13 provides an independent mechanism for the assessment of methods of protein structure modeling. From May through July 2018, CASP organizers had been posting on this website sequences of unknown protein structures for modeling. Protein models were collected from May through mid-August, and evaluated as the experimental coordinates became available. In the Summer and Fall, the tens of thousands of models submitted by approximately 100 research groups worldwide were processed and evaluated. Independent assessors in each of the prediction categories bring objectivity, balance, and independent insight to this process. Tools for viewing, comparison, and analysis of submitted models are available at this site. The results of the CASP13 experiment are first made public immediately before the Conference in December.
The results are currently getting published in a scientific journal Proteins.
The next round of CASP is planned to start in Spring 2020.

Targets

- Target List
- Domain Definition

Data guide

- CASP13 prediction format
- NMR restraint format (G.Montelione)
- NMR data tutorial (G.Montelione)
- SAXS data format (S.Tsutakawa, G.Hura)
- XLMS data tutorial (A.Leitner, E.Trabjerg)
- SANS data tutorial (A.Martel, S.Grudinin)
- FRET data format (C.Seidel)
- Energy Landscape Intro (G.Montelione)
- FRET data tutorial (C.Seidel)

Meeting

- Abstracts
- Program
- Groups Info
- Presentations

Results

AUTOMATIC EVALUATION

CASP13 results will be published
in a scientific journal Proteins in 2019.

Parseable Data

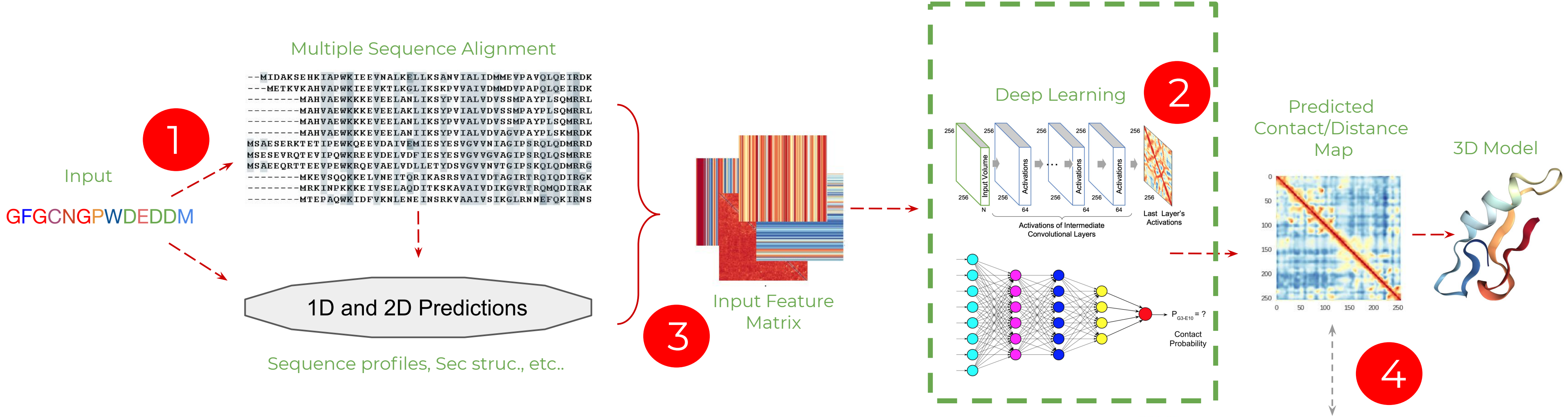
- Rankings: **Regular targets** (T)
- Rankings: **Refinement targets** (R)
- Rankings: **Data-Assisted targets** (S,X,N,A)
- Rankings: **Multimeric targets** (H,To)
- Rankings: **Contact predictions**

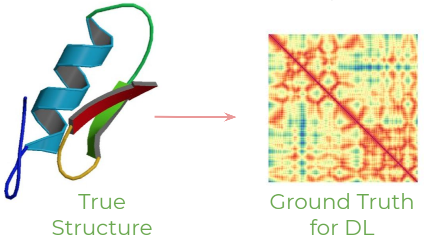
CASP13 in news and blogs

- Science
- The Guardian
- Forbes
- Deep Mind blog
- M.Alquraishi blog

Contact/distance prediction holds a key to solve the protein folding problem!

Attacking the Protein Contact Prediction Problem





1. Generating Large and **High-quality MSAs**

DL

2. The Right Deep Learning **Architecture**
3. **Feature Engineering** (encoding, 1D to 2D, covariance/precision matrix)
4. Appropriate **Loss Function** (besides binary_crossentropy, MSE, etc.)

What DL Method is Best Fit for Contact (or distance) Prediction?

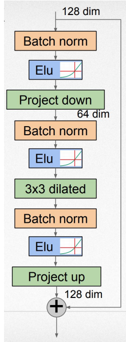
Top methods in the most recent CASP 13 Experiment

De novo protein folding using statistical potentials from deep learning

R.Evans, J.Jumper, J.Kirkpatrick, L.Sifre, T.F.G.Green, C.Qin, A.Zidek, A.Nelson, A.Bridgland, H.Penedones, S.Petersen, K.Simonyan, D.T.Jones [UCL], K.Kavukcuoglu, D.Hassabis, A.W.Senior

DeepMind

Group 043 / A7D / AlphaFold



DeepMetaPSICOV (DMP) in CASP13

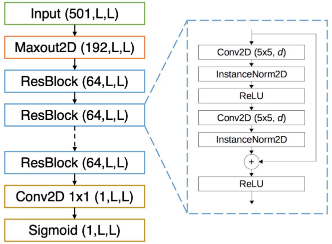
Shaun M Kandathil
University College London
&
The Francis Crick Institute

DeepMetaPSICOV model architecture

Deep, fully convolutional residual network

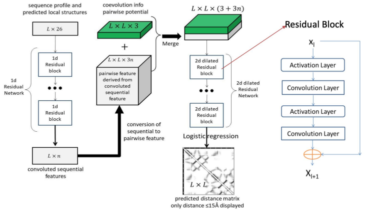
Total of 18 residual blocks plus one Maxout layer

Dilated convolutions



Distance-based Protein Folding Powered by Deep Learning
`

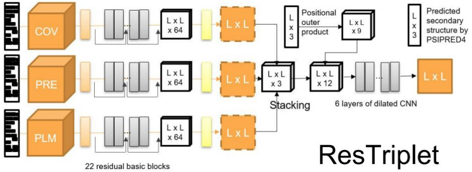
Jinbo Xu
Toyota Technological Institute at Chicago
(affiliate of Univ. of Chicago)



ResTriplet/TripletRes:
Learning contact-maps from a triplet of coevolutionary matrices

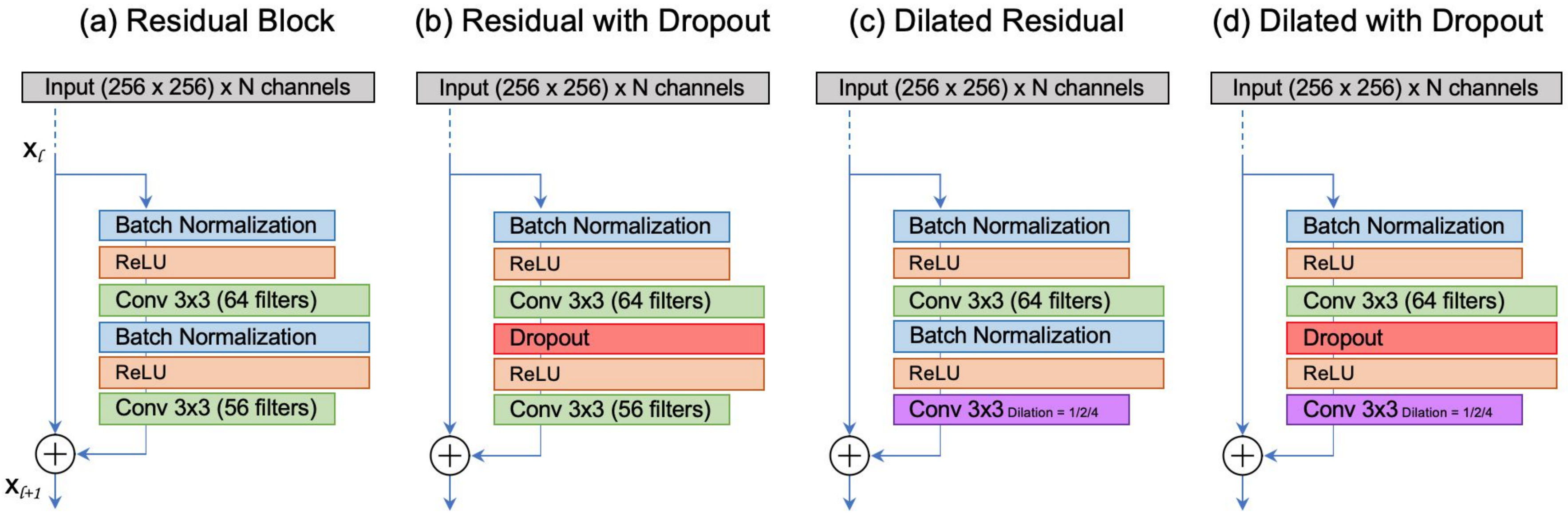
Eric W. Bell, Yang Li, Chengxin Zhang,
Dong-Jun Yu, Yang Zhang

Department of Computational Medicine and Bioinformatics,
University of Michigan - Ann Arbor

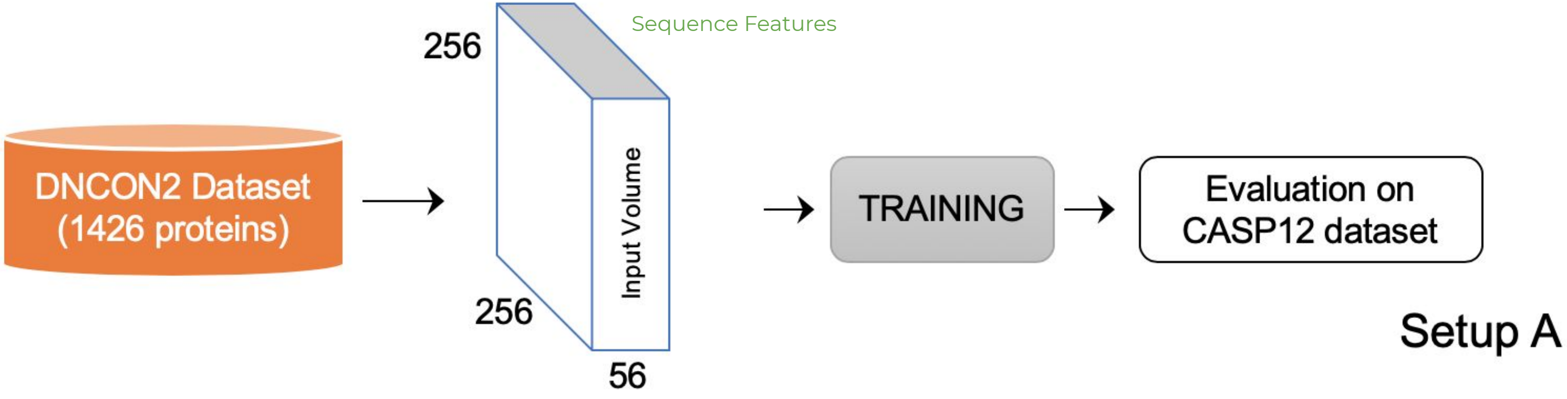


All these results show that residual networks are best architectures (for this problem)

What Variations of Residual Architectures?

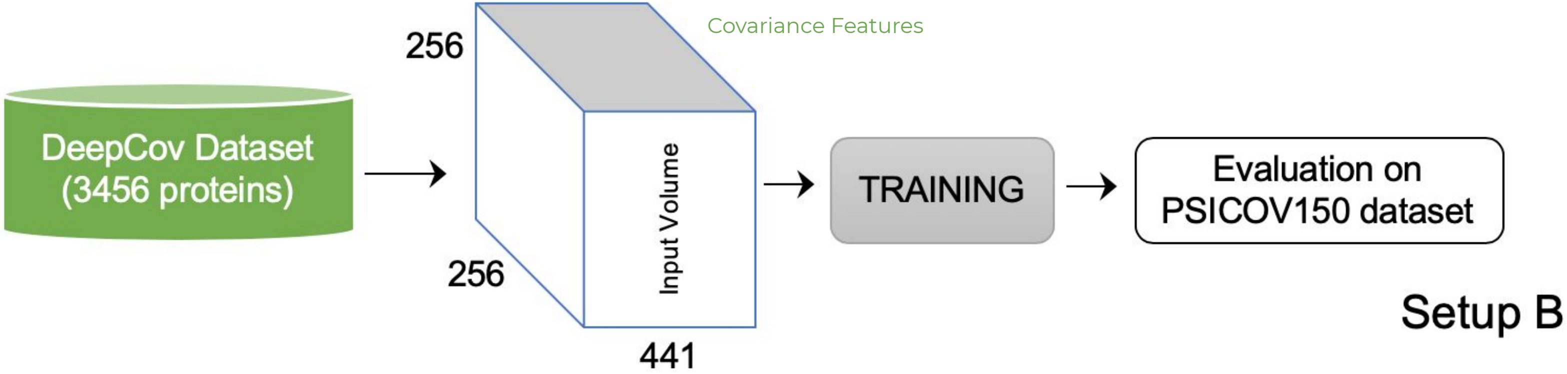


Experimental Setup



DNCON2 Dataset (1426 proteins) → Input Volume → TRAINING → Evaluation on CASP12 dataset

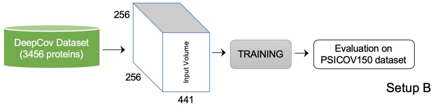
Setup A



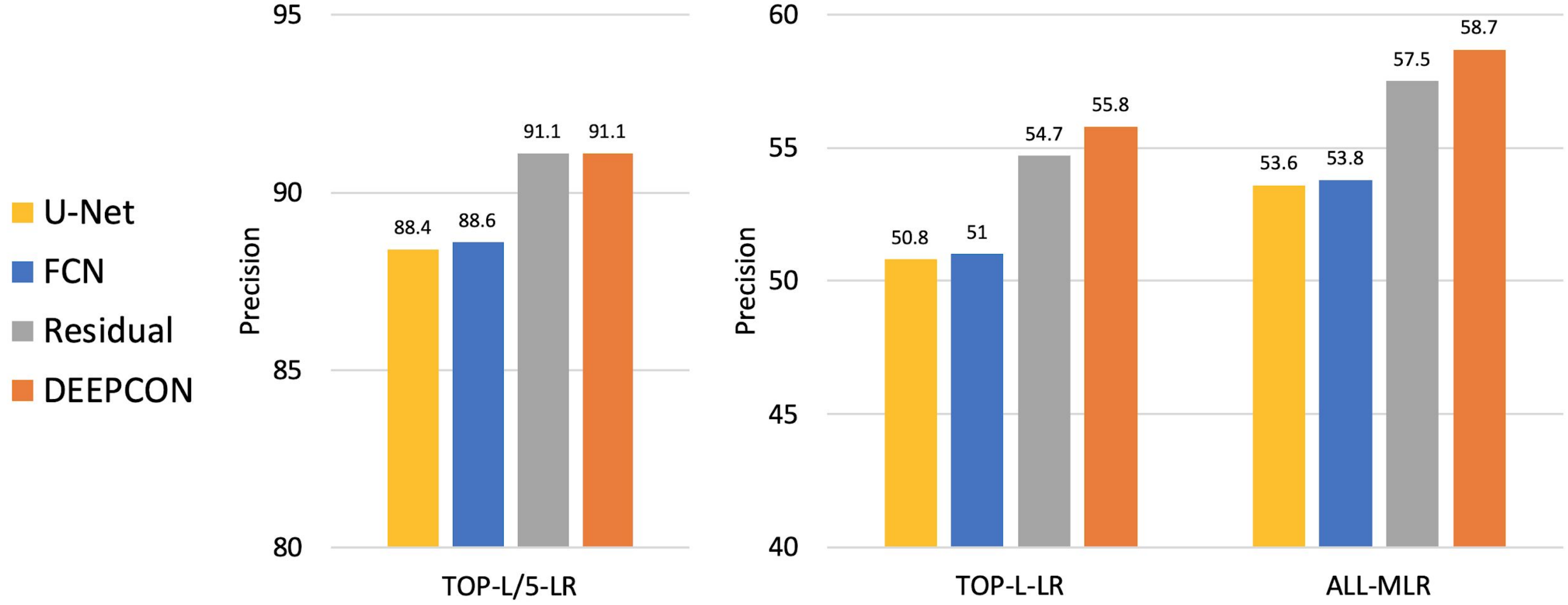
DeepCov Dataset (3456 proteins) → Input Volume → TRAINING → Evaluation on PSICOV150 dataset

Setup B

With Covariance Features as Input

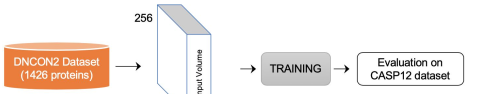


Setup B

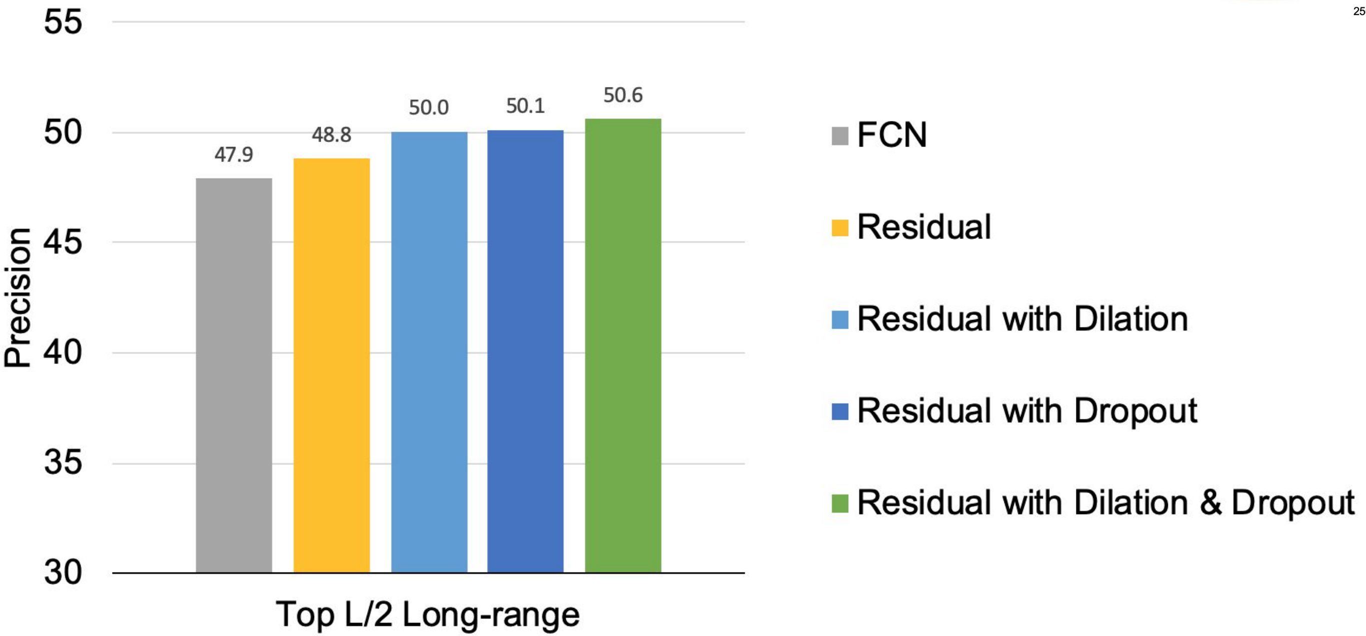


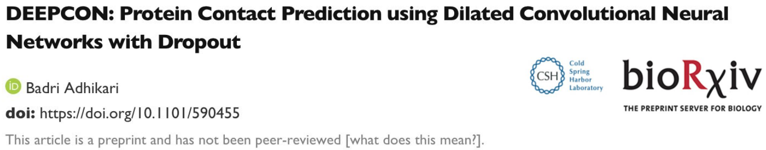
Fully connected ConvNets (FCN), residual networks with dropout and dilation (DEEPCON), regular residual network, and U-Net like architecture – using $P_{L/5\text{-}LR}$, $P_{L\text{-}LR}$, and $P_{NC\text{-}MLR}$. Trained and validated using the DeepCov dataset consisting of 3,456 proteins and tested using the PSICOV 150 dataset using the covariance features (left).

With Sequence Features as Input


DNCON2 Dataset (1426 proteins)
256
Input Volume
256
56
TRAINING
Evaluation on CASP12 dataset

Setup A




DEEPCON: Protein Contact Prediction using Dilated Convolutional Neural Networks with Dropout

Badri Adhikari

doi: https://doi.org/10.1101/590455

This article is a preprint and has not been peer-reviewed [what does this mean?].

DEEPCON is not a #1 Contact Predictor!

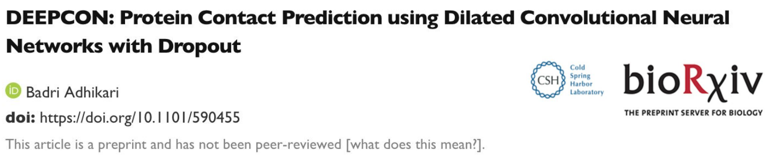
	$\mathbf{P_{L/5\text{-}LR}}$	$\mathbf{P_{L\text{-}LR}}$
* Best Group in CASP13 (RaptorX)	79.6	55.2
* RaptorX Webserver	67.4	43.8
DEEPCON (our method)	**57.2**	**34.2**
DeepCov (Jones and Kandathil, 2018)	48.2	28.3
PconsC4 (Michel *et al.*, 2018)	42.7	27.3
CCMpred (Seemayer *et al.*, 2014)	32.8	19.3
FreeContact (Kaján *et al.*, 2014)	30.3	17.4

Same Input Alignments

Ensembled + Larger Training set
+ Many other features

One Model + Small Training set
+ MSA Matrix as the only input

Comparison of DEEPCON's performance with other methods that accept multiple sequence alignment as input on the 32 domains in the CASP13 dataset of 20 protein targets. Methods with an asterisk (*) are listed for reference. All methods except for the first (in the first row) use the same alignment file as input.



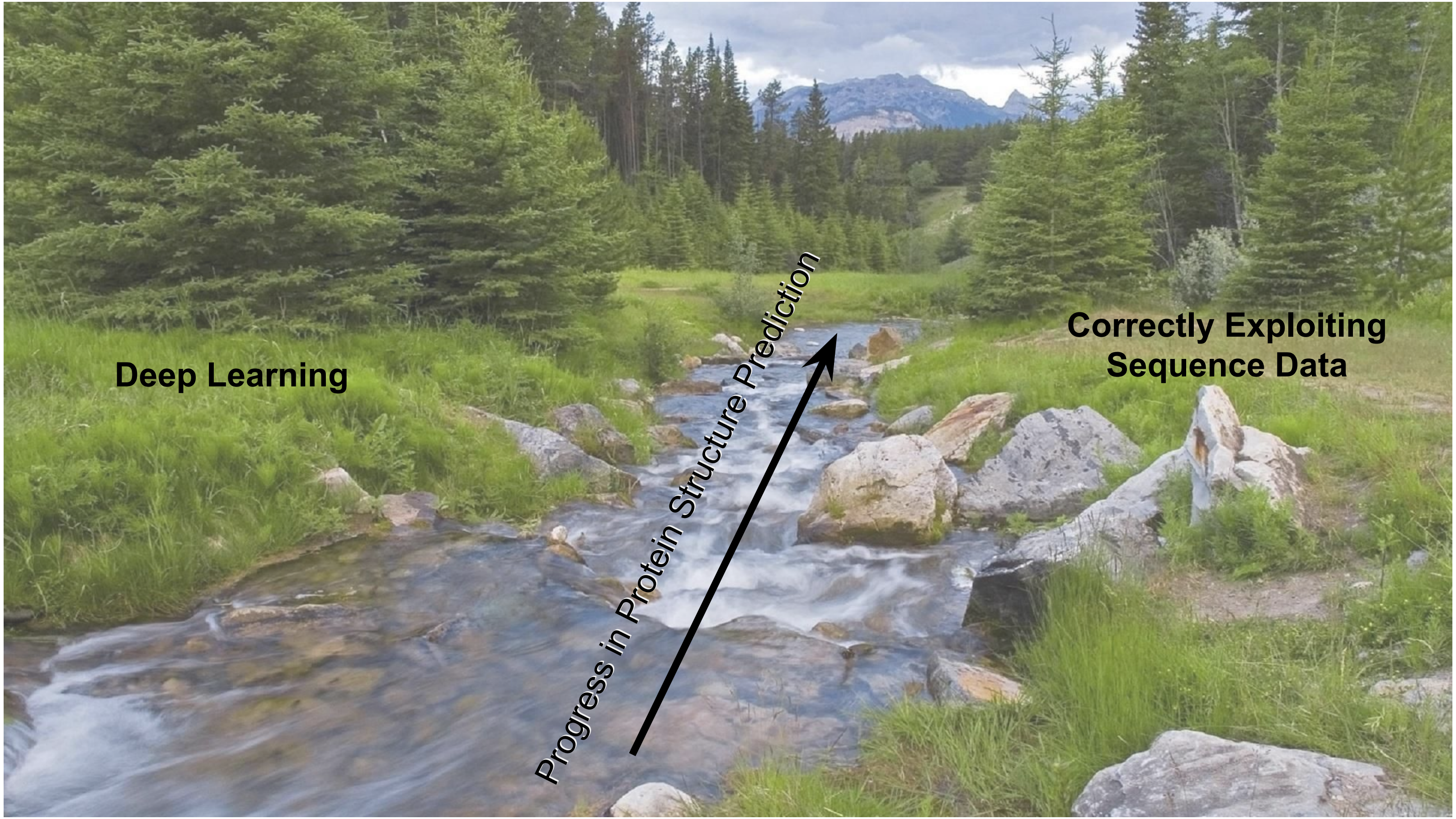
DEEPCON: Protein Contact Prediction using Dilated Convolutional Neural Networks with Dropout

Badri Adhikari

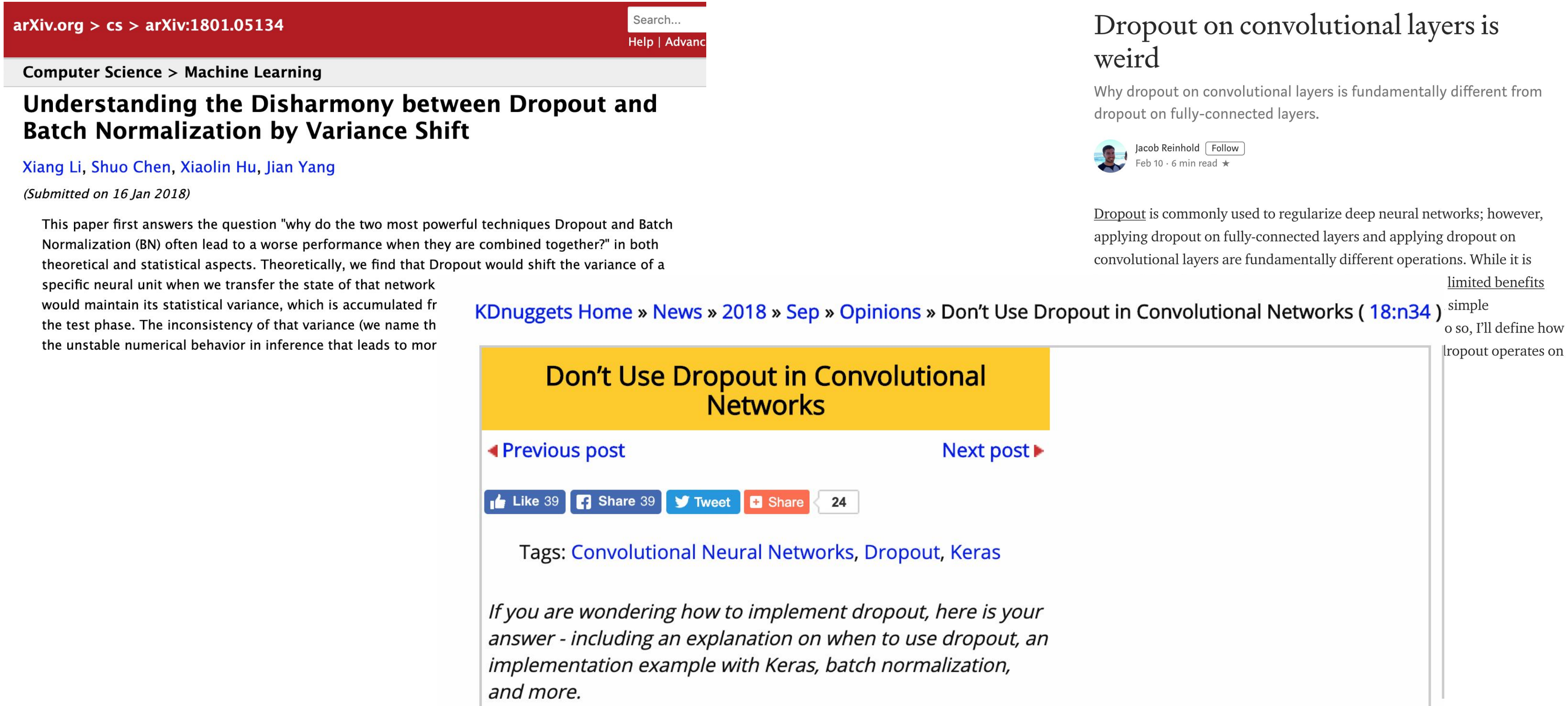
doi: https://doi.org/10.1101/590455

This article is a preprint and has not been peer-reviewed [what does this mean?].

Is There Room for Improvement?



An Example - Does Dropout really work with CNNs?



arXiv.org > cs > arXiv:1801.05134

Computer Science > Machine Learning

Understanding the Disharmony between Dropout and Batch Normalization by Variance Shift

Xiang Li, Shuo Chen, Xiaolin Hu, Jian Yang

(Submitted on 16 Jan 2018)

This paper first answers the question "why do the two most powerful techniques Dropout and Batch Normalization (BN) often lead to a worse performance when they are combined together?" in both theoretical and statistical aspects. Theoretically, we find that Dropout would shift the variance of a specific neural unit when we transfer the state of that network
would maintain its statistical variance, which is accumulated fr
the test phase. The inconsistency of that variance (we name th
the unstable numerical behavior in inference that leads to mor

Dropout on convolutional layers is weird

Why dropout on convolutional layers is fundamentally different from dropout on fully-connected layers.

Jacob Reinhold
Feb 10 · 6 min read ★

Dropout is commonly used to regularize deep neural networks; however, applying dropout on fully-connected layers and applying dropout on convolutional layers are fundamentally different operations. While it is
limited benefits
simple
o so, I'll define how
ropout operates on

KDnuggets Home » News » 2018 » Sep » Opinions » Don't Use Dropout in Convolutional Networks (18:n34)

Don't Use Dropout in Convolutional Networks

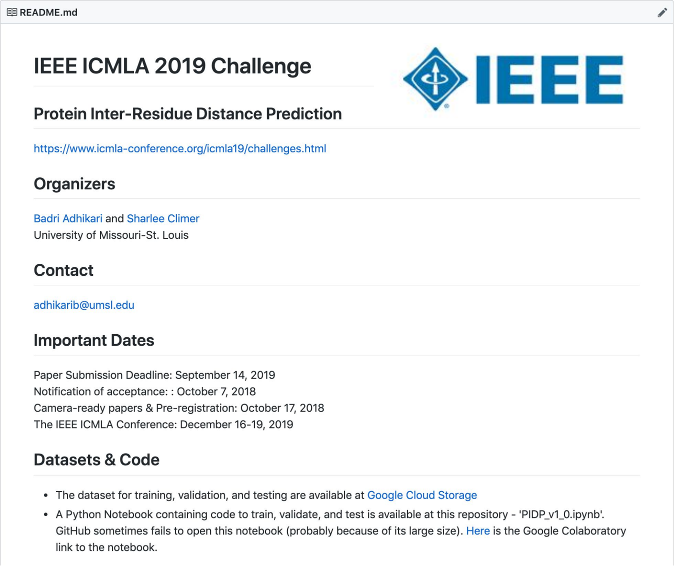
Tags: Convolutional Neural Networks, Dropout, Keras

If you are wondering how to implement dropout, here is your answer - including an explanation on when to use dropout, an implementation example with Keras, batch normalization, and more.

Conclusions

- Residual networks are NOT the “final” architectures for predicting protein contacts
    - The results in this work is an example
    - At least something like “NASNet” for computer vision must be done
    - Methods such as Capsule Networks could hold a lot of promise
- Architectures that work well for image datasets may not work well for protein contact prediction
    - U-Net architecture (~millions of parameters) performs worse than FCN (less than a million params)
- Protein contact prediction is now a problem for both (at least) - machine learning experts and bioinformaticians

Protein Distance Prediction Challenge



IEEE ICMLA 2019 Challenge

Protein Inter-Residue Distance Prediction

https://www.icmla-conference.org/icmla19/challenges.html

Organizers

Badri Adhikari and Sharlee Climer
University of Missouri-St. Louis

Contact

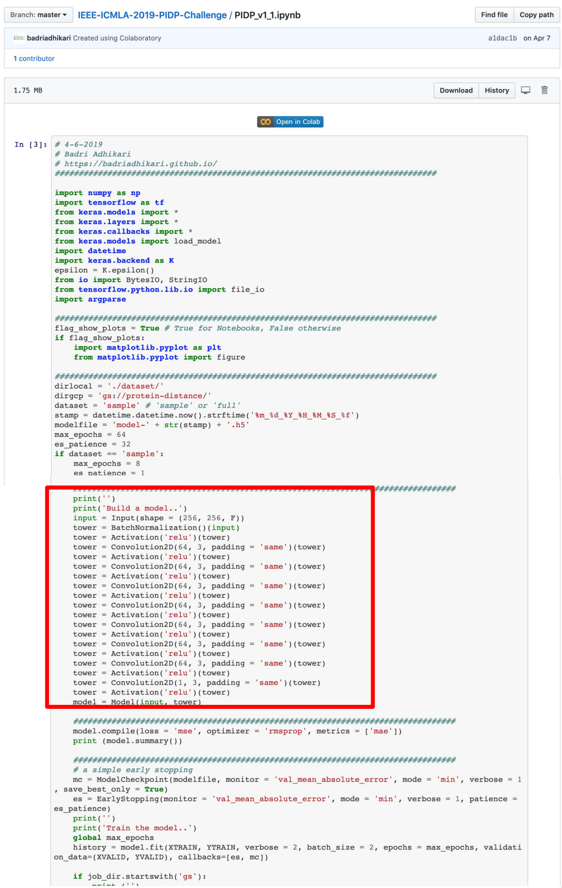
adhikarib@umsl.edu

Important Dates

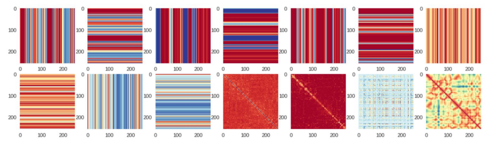
Paper Submission Deadline: September 14, 2019
Notification of acceptance: : October 7, 2018
Camera-ready papers & Pre-registration: October 17, 2018
The IEEE ICMLA Conference: December 16-19, 2019

Datasets & Code

- The dataset for training, validation, and testing are available at Google Cloud Storage
- A Python Notebook containing code to train, validate, and test is available at this repository - 'PIDP_v1_0.ipynb'. GitHub sometimes fails to open this notebook (probably because of its large size). Here is the Google Colaboratory link to the notebook.



```
# 4-6-2019
# Badri Adhikari
# https://badriadhikari.github.io/
################################################################################

import numpy as np
import tensorflow as tf
from keras.models import *
from keras.layers import *
from keras.callbacks import *
from keras.models import load_model
import datetime
import keras.backend as K
epsilon = K.epsilon()
from io import BytesIO, StringIO
from tensorflow.python.lib.io import file_io
import argparse

################################################################################
flag_show_plots = True # True for Notebooks, False otherwise
if flag_show_plots:
    import matplotlib.pyplot as plt
    from matplotlib.pyplot import figure

################################################################################
dirlocal = './dataset/'
dirgcp = 'gs://protein-distance/'
dataset = 'sample' # 'sample' or 'full'
stamp = datetime.datetime.now().strftime('%m_%d_%Y_%H_%M_%S_%f')
modelfile = 'model-' + str(stamp) + '.h5'
max_epochs = 64
es_patience = 32
if dataset == 'sample':
    max_epochs = 8
    es_patience = 1

    ############################################################################
    print('')
    print('Build a model..')
    input = Input(shape = (256, 256, F))
    tower = BatchNormalization()(input)
    tower = Activation('relu')(tower)
    tower = Convolution2D(64, 3, padding = 'same')(tower)
    tower = Activation('relu')(tower)
    tower = Convolution2D(64, 3, padding = 'same')(tower)
    tower = Activation('relu')(tower)
    tower = Convolution2D(64, 3, padding = 'same')(tower)
    tower = Activation('relu')(tower)
    tower = Convolution2D(64, 3, padding = 'same')(tower)
    tower = Activation('relu')(tower)
    tower = Convolution2D(64, 3, padding = 'same')(tower)
    tower = Activation('relu')(tower)
    tower = Convolution2D(64, 3, padding = 'same')(tower)
    tower = Activation('relu')(tower)
    tower = Convolution2D(64, 3, padding = 'same')(tower)
    tower = Activation('relu')(tower)
    tower = Convolution2D(1, 3, padding = 'same')(tower)
    tower = Activation('relu')(tower)
    model = Model(input, tower)

    ############################################################################
    model.compile(loss = 'mse', optimizer = 'rmsprop', metrics = ['mae'])
    print (model.summary())

    ############################################################################
    # a simple early stopping
    mc = ModelCheckpoint(modelfile, monitor = 'val_mean_absolute_error', mode = 'min', verbose = 1
, save_best_only = True)
    es = EarlyStopping(monitor = 'val_mean_absolute_error', mode = 'min', verbose = 1, patience =
es_patience)
    print('')
    print('Train the model..')
    global max_epochs
    history = model.fit(XTRAIN, YTRAIN, verbose = 2, batch_size = 2, epochs = max_epochs, validati
on_data=(XVALID, YVALID), callbacks=[es, mc])

    if job_dir.startswith('gs'):
```



Acknowledgements

Research Support & Contribution


Cezary Janikow


Sharlee Climer


Cynthia Jobe


Anthony A.-Nyanzu

Chase Richards


Patrick Kong


Sri Harsha Akurathi

IT Support

Philip Reiss

Kenneth Voss

Michael Remier

MU - Research Computing Support Services (RCSS)

Computing Resources

Google

NVIDIA

University of Missouri System

COLUMBIA | KANSAS CITY | ROLLA | ST.LOUIS