Bare Demo of IEEERev.cls for USNC-URSI National Radio Science Meeting

Authors Name/s associated with 1st Affiliation
line 1: dept. name (if applicable)
line 2: name of organization, acronyms acceptable
line 3: City, State/Province, Country
line 4: e-mail address if desired

Authors Name/s associated with 2nd Affiliation
line 1: dept. name (if applicable)
line 2: name of organization, acronyms acceptable
line 3: City, State/Province, Country
9444444444444787 line 4: e-mail address if desired

***Abstract*—The abstract goes here.**

***Index Terms*—IEEE, IEEEtran, journal, LaTeX, paper, template.**

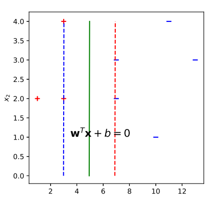
I. Show Homework

sadfasf, sdfdsf, sdf.
Test citations:
[1] [2] [3].

A. Show Floats

Test figures and example block which is shown in Example I.1.

Example I.1 (Figure Problem) *Test inner subgraphs, i.e.* Fig. 1(a) and Fig. 1(b). Also test 1(b) and (1-1):



(a) $D = 1$

Here could be graphs.

(b) $D = 0.5$

Fig. 1. Test graphs.

□

Test subequations and the theorem block which is shown in Theorem I.1.

Theorem I.1 (Example Theorem) *Here we show a simple example of subequations in* (1-1)*:*

$$\frac{\partial \mathcal{L}(\mathbf{w},\ b)}{\partial \mathbf{w}} = \mathbf{w} + C \sum_i \frac{\partial \ell_i}{\partial \mathbf{w}}, \tag{1-1}$$

$$\frac{\partial \mathcal{L}(\mathbf{w},\ b)}{\partial b} = C \sum_i \frac{\partial \ell_i}{\partial b}, \tag{1-2}$$

Test table, which is shown in Table I:
Test equations in (2):

TABLE I
Parameters of *Daubechies*'s filter.

n	$h[n]$	$g[n]$
0	0.3327	-0.0352
1	0.8069	-0.0854
2	0.4599	0.1350
3	-0.1350	0.4599
4	-0.0854	-0.8069
5	0.0352	0.3327

$$\begin{aligned} I(\Omega) &= \mathrm{Re}\left\{ \frac{e^{-x}}{j\Omega} e^{j\Omega x}\Big|_0^1 + o\left(\frac{1}{\Omega}\right) \right\} \approx \mathrm{Re}\left\{ \frac{e^{-x}}{j\Omega} e^{j\Omega x}\Big|_0^1 \right\} \\ &= \mathrm{Re}\left\{ \frac{e^{j\Omega - 1} - 1}{j\Omega} \right\} = \frac{1}{\Omega e} \cos\left(\Omega - \frac{\pi}{2}\right) = \frac{1}{\Omega e} \sin \Omega. \end{aligned} \tag{2}$$

B. Show Algorithm

Test Algorithm in Algorithm 1:

Algorithm 1 DWT Algorithm

Input: Sequence $\mathbf{x}$ in time domain
Output: Sequence $\hat{\mathbf{x}}$ in wavelet domain

N = $\lfloor \log_2(\text{length}(\mathbf{x})) \rfloor$;
$\mathbf{c}_N = \mathbf{x},\ \hat{\mathbf{x}} = \varnothing$;
for i from 1 to N **do**
  $\mathbf{c}_{N-i},\ \mathbf{d}_{N-i} = \text{analysis\_filter}(\mathbf{c}_{N-i+1})$;
  insert $\mathbf{d}_{N-i}$ at the beginning of $\hat{\mathbf{x}}$.
end for

Test codings:

```
# HyperPlate of SVM. It contains variables
    including w and b, and convert input x
    vector to a single value y(+-1).
with tf.name_scope('SVMPlate'): #Noted that the
    dimension of y must be 1, so the constants
    should be 1 dimensional.
    self.constrain = tf.constant(
        SVMPrimalSolution.Domain, dtype=tf.
        float32, shape=[1], name='Constrain')
    self.w = self.weight_variable([1, self.xDim],
         name='Weight')
    bias = self.bias_variable([1], name='Bias')
    self.subjection = tf.multiply(self.y, tf.
        matmul(self.w, self.x) + bias)
```

```python
    tf.add_to_collection('Weight', self.w)
    tf.add_to_collection('Bias', bias)

@staticmethod
def weight_variable(shape, name=None):
    '''weight_variable generates a weight variable of a given shape.'''
    initial = tf.truncated_normal(shape, stddev=0.1)
    if name is not None:
        return tf.Variable(initial, name=name)
    else:
        return tf.Variable(initial)

@staticmethod
def bias_variable(shape, name=None):
    ''''bias_variable generates a bias variable of a given shape.'''
    initial = tf.constant(0.1, dtype=tf.float32, shape=shape)
    if name is not None:
        return tf.Variable(initial, name=name)
    else:
        return tf.Variable(initial)
```


Acknowledgment

The authors would like to thank...

II. References

[1] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 2528–2535.

[2] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467–3478, Aug 2012.

[3] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb 2016.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.