Diferenciación De Oradores Mediante Un Autocodificador Convolucional

Resumen: en este trabajo, una solución de aprendizaje profundo para diferenciar las voces de los altavoces en el audio dadas las dos fuentes de micrófono se presenta como un paso hacia la solución del problema de la fiesta de cóctel. Se entrenó a un autocodificador convolucional utilizando un pequeño tamaño de muestra de datos para asociar fragmentos de audio con etiquetas categóricas. Los fragmentos de audio recopilados como parte de este trabajo se utilizaron para entrenar y evaluar el modelo. El audio se convirtió en

representación de cepstrum de frecuencia mel antes de la clasificación. Los datos procesados colectivamente se etiquetaron de acuerdo con la persona o la colección de personas que hablaban. El modelo se entrenó y evaluó utilizando datos de dos, tres, cuatro, cinco y seis categorías. El resultado fue un modelo que reconoce cuando diferentes personas están hablando en una conversación de 2 personas, 3 personas, 4 personas, 5 personas y 6 personas con una precisión del 99,29 %, 97,62 %, 96,43 %, 93,43 % y 88,1 %, respectivamente. Se presentan comparaciones experimentales entre las cinco versiones del modelo.

Términos de índice: aprendizaje profundo, red neuronal convolucional, separación de voz, procesamiento de señales

I. Introducción

Este documento describe la investigación,

el diseño y la implementación de una solución de aprendizaje profundo para diferenciar las voces de los altavoces en el audio procedente de dos micrófonos simultáneamente. El arte anterior, como [1], se centró en el uso de una solución de aprendizaje profundo para realizar la separación de altavoces dada una señal de audio mixta. Este trabajo se centra en el uso de una solución de aprendizaje profundo para realizar la diferenciación o clasificación de los altavoces dadas las señales de audio en las que un solo orador está hablando a la vez. El modelo desarrollado en este trabajo se presenta como un primer paso para resolver el problema del cóctel.

El objetivo de este trabajo era evaluar cómo funcionaría un modelo de aprendizaje automático para resolver la diferencia de altavoces en audio cuando el número de oradores y el tamaño del conjunto de datos era mayor o menor.

Bucketization, que se conoce comúnmente como binning multivariado o Análisis de datos multivariados, es el proceso de extraer solo características relevantes de los datos dados para clasificarlos [2]. En este trabajo, se extrajeron características relevantes de los fragmentos de audio para clasificar los fragmentos para que pertenecieran a un orador específico (altavoz). En los datos multidimensionales, no todas las características son relevantes a la hora de clasificar las muestras de datos [2]. Como tal, es necesario un enfoque en el que solo queden características interesantes [2]. Se han utilizado métodos tradicionales como el análisis de componentes principales (PCA) y la escala multidimensional (MDS) para la extracción de características relevantes dados datos multidimensionales [2]. Este trabajo describe el uso de una solución de aprendizaje automático para realizar la extracción de características de per-form,

así como la bucketización automáticamente. Se han implementado muchos tipos diferentes de modelos de aprendizaje automático para resolver varios problemas y desafíos relacionados con las señales de audio. El modelo de aprendizaje automático utilizado en este trabajo fue un autocodificador convolucional (CAE).

II. Fondo

Los autocodificadores (AE) se componen de dos partes, un codificador y un decodificador. El codificador reducirá las dimensiones de la entrada a una representación de espacio latente, mientras que el decodificador intentará reconstruir los datos comprimidos [3]. El muestreo descendente se utiliza durante la etapa de codificador y el muestreo descendente durante la etapa de decodificador para comprimir y reconstruir los datos de entrada, respectivamente [4].

Los AE, específicamente los CAE, se han utilizado en muchas aplicaciones que implican el uso de datos de imagen. Los CAE han sido efectivos en la extracción de características visuales de las imágenes de entrada y, como resultado, han permitido la generación de representaciones de espacio latente mucho más precisas en comparación con los AE tradicionales [4]. La figura 1 proporciona una representación visual de la estructura de un AE.

Los CAE se basan en AE estándar donde se utilizan capas convolucionales para la codificación y/o decodificación [4]. CAEs preservan

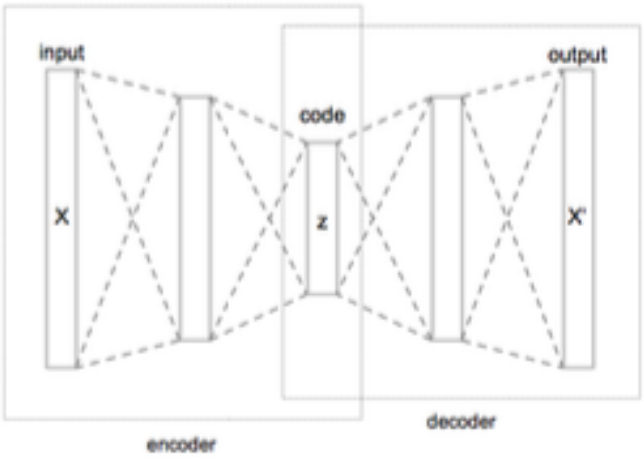

Fig. 1: Autoencoder Structure

Localidad espacial, ya que los pesos se comparten entre todas las ubicaciones de entrada [4]. Durante el entrenamiento, se utiliza la propagación posterior para calcular el gradiente de la función de error. Esto permite actualizar los pesos de la red utilizando un algoritmo de descenso de gradiente como Adadelta [4].

III. ARTE ANTERIOR

Las representaciones de cepstrum de frecuencia Mel (MFCC) han dominado el campo del reconocimiento de voz en términos de representación de

características, ya que proporcionan una forma compacta de la representación del espectro de amplitud del audio [5]. Los MFCC proporcionan una representación espectral a corto plazo de las características de audio [5]. El trabajo en [5] encontró que los MFCC no tenían un efecto negativo en los algoritmos de discriminación del habla y la música cuando se usaban como paso de preprocesamiento para la extracción de características.

Los CAE se han utilizado como una solución a muchos problemas diferentes que van desde la clasificación hasta el desnido. Un ejemplo de aplicación de CAE en datos de desnudo, es en el campo de la desnudo de imágenes médicas [6]. El trabajo en [6] describe cómo el uso de datos de tamaño de muestra pequeño junto con un CAE puede desañar de manera eficiente las imágenes médicas.

Los CAE también se han aplicado a los problemas basados en la clasificación como se describe en [7]. En [7], se utilizó un CAE para clasificar imágenes de radar de apertura sintética (SAR) de alta resolución que están contaminadas por el ruido de moteado [7]. El trabajo en [7] describe cómo los CAE son capaces de extraer características, así como de clasificar las imágenes SAR automáticamente, eliminando la necesidad de una extracción de características manual y que consume mucho tiempo.

El campo de la separación y clasificación de oradores también ha hecho uso de algoritmos de aprendizaje automático [8]. En [8], el objetivo era resolver el problema del cóctel con técnicas y algoritmos modernos de aprendizaje automático. Se utilizó un clasificador de red neuronal de regresión general como solución para el

problema del cóctel [8]. En [8] se utilizó una red neuronal artificial multicapa para clasificar el enfoque de atención de un oyente en un entorno de varios altavoces, dadas las características extraídas de los datos del electroencefalograma. El modelo se evaluó en función de la precisión de la clasificación, la sensibilidad, la especificidad y el tiempo de cálculo. Los resultados fueron de alrededor del 99 % de precisión de clasificación,

98,9 % de sensibilidad, 99,1 % de especificidad y alrededor de 8 segundos de tiempo de computación [8].

Otro ejemplo de aplicación de una solución de aprendizaje automático en la separación de altavoces se describe en [9]. Al igual que [8], en [9] se propuso una solución moderna de aprendizaje automático para resolver el problema del cóctel. En [9], se propuso una solución en

la que se utilizó la invarianza y el entrenamiento de la permutación a nivel de expresión para entrenar una red neuronal recurrente de memoria bidireccional a largo plazo [9]. El modelo resultante fue capaz de mejorar la relación señal-distorsión, así como la inteligibilidad objetiva extendida a corto plazo para las desafiantes relaciones señal-ruido en el audio de varios altavoces [9].

IV. SISTEMA DESARROLLADO

En el estado de la técnica (por ejemplo, [8], [9]), se implementaron soluciones de aprendizaje automático no CAE para resolver la separación de altavoces en el audio de varios altavoces. Sin embargo, los sistemas desarrollados requerían un preprocesamiento de audio pesado, grandes conjuntos de datos y demostraron ser computacionalmente intensivos. Este trabajo describe el uso de

una solución moderna de aprendizaje automático para resolver la diferenciación de altavoces en el audio de varios altavoces. Se diseñó y desarrolló una solución que requiere un preprocesamiento de audio mínimo, conjuntos de datos de tamaño de muestra pequeño y que era computacionalmente eficiente. Se entrenó y evaluó un modelo CAE en el origen de audio a partir de dos micrófonos simultáneamente. El objetivo era evaluar la precisión del CAE a medida que aumenta el número de cubos en la salida de la red. La expectativa era que una combinación específica de calidad del conjunto de datos, tamaño del conjunto de datos y tamaño del cubo proporcionaría la mejor precisión, recuperación y precisión por parte del CAE para diferenciar los altavoces en audio. A medida que aumente el tamaño del cubo en la salida del CAE, es decir, aumentar el número de neuronas de salida, la matriz de confusión resultante

será más grande. Con N cubos, se construiría una matriz N x N. La expectativa era que, sin importar el tamaño y la calidad del conjunto de datos, a medida que aumentara el número de cubos (es decir, N se hace más grande), la precisión del modelo eventualmente disminuiría.

En el estado de la técnica (por ejemplo, [6], [7]), la parte del decodificador del CAE se utilizó para reconstruir los datos reducidos a la representación del espacio latente. Sin embargo, en este trabajo, el decodificador del CAE utilizó las características relevantes en los datos codificados, los datos que se proporcionaron como entrada al decodificador, con el fin de diferenciar los altavoces en el audio original colocándolos dentro de cubos. Como tal, el decodificador en este trabajo era simplemente una red neuronal densa (DNN) en lugar de una convolucional en

la que se proporcionaban los datos codificados aplanados como entrada.

La estructura del CAE implementada en este trabajo es bastante sencilla. El codificador estaba compuesto por una capa de entrada seguida de tres capas convolucionales, una capa de agrupación y una capa de caída. El método de agrupación más utilizado en las soluciones modernas de aprendizaje automático, Max-Pooling, se utilizó para realizar la reducción de la dimensionalidad [10]. La reducción de la dimensión se realiza comúnmente en la etapa de preprocesamiento de las aplicaciones de agrupación y clasificación [11]. En este trabajo, la reducción de la dimensionalidad fue parte de la etapa de análisis de datos en lugar de la etapa de preprocesamiento. La salida reducida del decodificador se aplanó con una capa aplanada. El la capa de aplanamiento permite aplanar los mapas de

características de la salida del decodificador combinándolos [12]. Los datos aplanados se proporcionaron entonces como entrada al decodificador, que estaba compuesto por capas densas con capas de caída en el medio. Se utilizaron capas de caída, ya que ayudan a evitar que el modelo se sobreajuste [13]. La estructura del CAE implementado en este trabajo se describe con más detalle en la Tabla I y la Figura 2.

TABLE I: CAE For Differentiating 2 Speakers

Layer	Output Shape	Parameters
Input Layer	(40, 11, 1)	0
Conv2D Layer	(19, 10, 32)	160
Conv2D Layer	(18, 9, 48)	6192
Conv2D Layer	(17, 8, 120)	23160
MaxPooling2D Layer	(8, 4, 120)	0
Dropout Layer	(8, 4, 120)	0
Flatten Layer	(3840)	0
Dense Layer	(128)	491648
Dropout Layer	(128)	0
Dense Layer	(64)	8256
Dropout Layer	(64)	0
Dense Layer	(2)	130

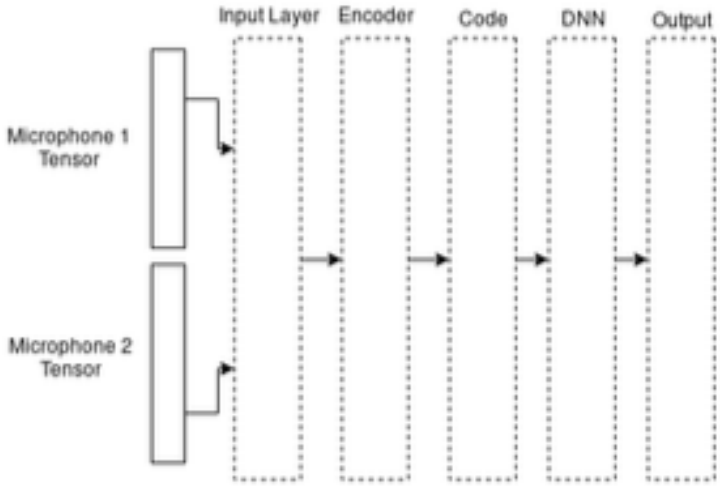

Fig. 2: Model Diagram

La función de activación utilizada para todas las capas, aparte de la capa de salida, fue la unidad lineal rectificada (ReLU), ya que es eficiente desde el punto de vista computacional [14]. La

función de activación utilizada para la capa de salida fue softmax, ya que el decodificador se utilizó para la clasificación de varias clases. La función de pérdida fue una entropía cruzada categórica con Adadelta como optimizador. Se eligieron la activación de Softmax y la función de penalización de entropía cruzada, ya que hay un emparejamiento natural entre ellos [15]. Adadelta se utilizó como algoritmo de optimización, ya que especificar una tasa de aprendizaje es innecesario, ya que se ha eliminado de la regla de actualización del algoritmo [16]. Como tal, el número de hiperparámetros se redujo, ya que el modelo requería menos ajuste manual de la tasa de aprendizaje.

Antes de entrenar y probar el modelo, los conjuntos de datos de audio se procesaron previamente de la siguiente manera. El trabajo en [5], demostrado Que los MFCC pueden proporcionar una

representación espectral a corto plazo de las características de audio sin afectar negativamente a los algoritmos de discriminación del habla y la música. Como tal, con el fin de mejorar la eficiencia computacional del sistema desarrollado en este trabajo sin una reducción en la precisión, los datos de entrada se procesaron previamente transformando cada fragmento de audio en una representación MFCC con una longitud de ventana de 2048 y una longitud de salto de 512. Para calcular los MFCC de los fragmentos de audio utilizados en este trabajo, se utilizó el paquete python Librosa. Librosa es un paquete de Python que se usa comúnmente en el campo de la música para recuperar información de los datos de audio [17]. La conversión de los datos de audio en representación MFCC dio como resultado datos 2D similares a la imagen que eran similares a los datos de [6] [7]. Para cada archivo de audio, el

preprocesamiento dio como resultado un tensor de 20 x 11. En este caso, la función MFCC de Librosa calculó 20 MFCC en 11 fotogramas. Cada par de representaciones tensoras MFCC de fragmentos de audio idénticos, una de cada micrófono, se concatenaron para servir como entrada al CAE desarrollado en este trabajo. Como tal, la forma de los datos de entrada para el modelo fue (40, 11).

V. PREPARACIÓN DE DATOS

Los datos de entrenamiento y prueba se recopilaron utilizando dos microteléfonos simultáneamente. Se recogieron fragmentos de audio de un total de 6 oradres menores de 30 años, 3 de los cuales eran hombres y 3 de los cuales eran mujeres. Los hablantes también eran una mezcla de hablantes nativos y no nativos de inglés. Durante la grabación, los altavoces se colocaron a una distancia

de 2 m de ambos micrófonos, con los dos micrófonos a una distancia de 1 m de distancia. Cada micrófono estaba conectado a una Raspberry Pi 3 Modelo V1.2 que se ejecutaba en Rasbian OS, donde se iniciaban, se detenían y guardaban las grabaciones. Esto permitió fragmentos de audio separados idénticos de cada micrófono.

A los oradores se les dieron las mismas transcripciones del conjunto de datos Common Voice de Mozilla. Cada orador lee la misma lista de oraciones, donde cada frase fue grabada por cada micrófono y guardada en un archivo WAV. Cada par de archivos WAV generados se consideró como una sola muestra en el conjunto de datos utilizado en este trabajo. Cada archivo WAV tenía una frecuencia de muestreo de 44100 Hz y una duración de 10 segundos con el nivel de ruido en la habitación a un promedio de 47dB durante las grabaciones. Se

recogieron un total de 82 grabaciones para cada altavoz de cada micrófono para el conjunto de datos de entrenamiento, y se recogieron un total de 12 grabaciones para cada altavoz de cada micrófono para el conjunto de datos de prueba. Ambos conjuntos de datos se organizaron y etiquetaron de acuerdo con el altavoz presente en el fragmento de audio y el micrófono de origen utilizado para grabar ese fragmento. Cada altavoz representaría un cubo en la salida del modelo.

VI. RESULTADOS

El modelo se entrenó con un tamaño de lote de 64 sobre 80 épocas, es decir, 80 pases completos del conjunto de datos de entrenamiento a través del CAE, utilizando un conjunto de datos de 82 fragmentos de audio por altavoz. El conjunto de datos de entrenamiento se dividió entre un subconjunto de

entrenamiento de 70 fragmentos de audio por altavoz y un subconjunto de 10 fragmentos de audio por altavoz para evaluar el modelo. La precisión del modelo se evaluó mediante la construcción de matrices de confusión donde se calcularon los valores de verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN). La precisión y el recuerdo se calcularon utilizando TP, FN y FP. La Tabla II proporciona el recuerdo, la precisión y la precisión de cada matriz de confusión.

TABLE II: Confusion Matrix Results

# of Buckets	Recall (%)	Precision (%)	Accuracy (%)
2	100	100	100
3	89.01	100	94.51
4	97.43	97.22	97.33
5	90.38	91.67	91.03
6	72.11	76.39	76.57

Table II demuestra claramente que a medida que aumenta el número de dólares, la precisión del modelo disminuye. El siguiente diagrama

proporciona una visualización de la disminución de la precisión del modelo a medida que aumenta el número de cubos.

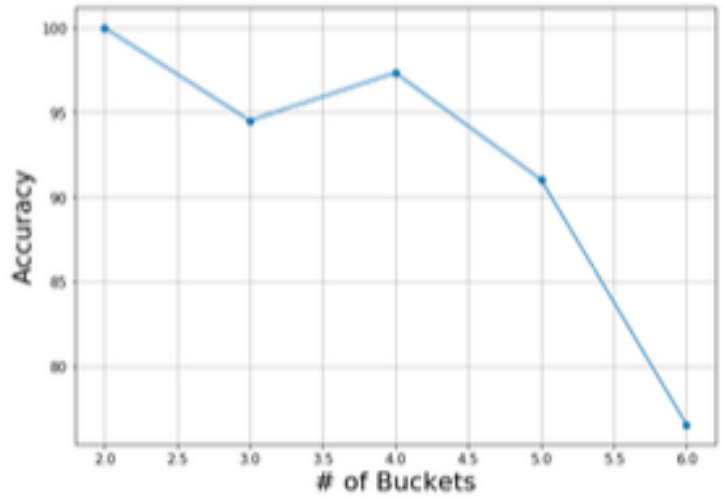

Fig. 3: Model Accuracy vs. Number of Speakers Classes

Los resultados extraídos de las matrices de confusión generadas indican que un CAE puede diferenciar los altavoces con audio procedente de dos micrófonos simultáneamente. Los resultados también indican que cuanto mayor sea el número de cubos en los que se entrenó el modelo, menos preciso era el modelo. Como tal, se ha confirmado la hipótesis de que a medida que aumenta el número de cubos, la precisión del modelo eventualmente disminuiría. Sin en modo, se debe tener en cuenta el número

limitado de muestras de prueba. La calidad y la cantidad de los datos tienen un impacto directo en el rendimiento de los algoritmos de aprendizaje automático [18]. Como tal, con el fin de verificar los resultados extraídos de las matrices de confusión, también se utilizó la validación cruzada de k-fold para evaluar el modelo. En este trabajo, el modelo se evaluó utilizando 10 pliegues, donde el rendimiento del modelo se evaluó en el pliegue de validación sostenido [19]. La tabla III y la figura 4 proporcionan los resultados de validación cruzada de k veces.

La Tabla III y la Figura 4 demuestran además que un CAE puede diferenciar los altavoces en el audio. Sin embargo, como se señaló anteriormente en base a los resultados de las matrices de confusión, la Tabla III y la Figura 4 también demuestran que como el número de

TABLE III: K-Fold Cross Validation Results – 2 Microphones

# of Buckets	Mean Accuracy (%)	Standard Deviation (%)
2	99.29	2.14
3	97.62	3.19
4	96.43	3.19
5	93.43	3.39
6	88.10	2.82

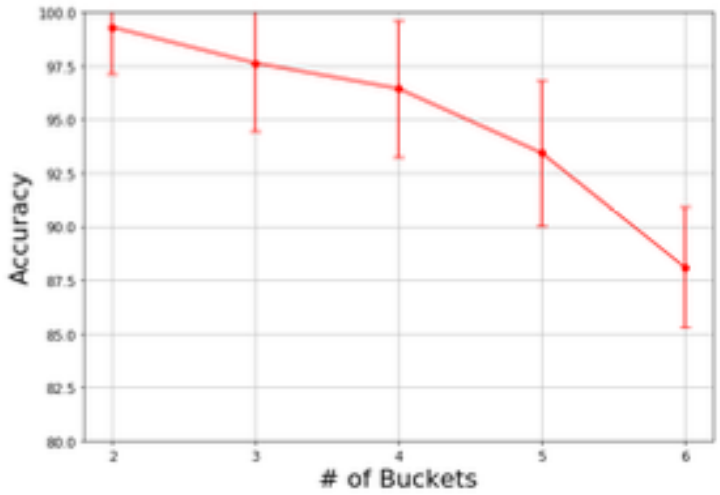

Fig. 4: KFold Cross Validation Results – 2 Microphones

oradores aumenta, la precisión del modelo disminuye. Los resultados obtenidos mediante la validación cruzada de k-fold también parecen ser ligeramente mejores que los obtenidos con matrices de confusión. En la validación cruzada de k veces, el conjunto de datos de entrenamiento se divide al azar en k subconjuntos, o pliegues, donde el modelo se entrena en cada subconjunto

excepto en uno. El rendimiento del modelo se evalúa en el subconjunto de validación sostenido [19]. En este trabajo, el modelo se probó en 10 pliegues a partir de un conjunto de datos de 82 muestras. El modelo se entrenó 10 veces y se evaluó cada vez en el subconjunto o pliegue aleatorio. A continuación, se evaluó el rendimiento del modelo tomando el promedio de los resultados de precisión de cada pliegue de evaluación. Como tal, dado el conjunto de datos limitado utilizado en este trabajo, la validación cruzada de k-fold proporciona una menor varianza y reduce el sesgo en el subconjunto de prueba en comparación con un solo conjunto de retención al evaluar el modelo.

En el preprocesamiento de datos, los tensores resultantes de cada mi-crophone se concatenaron y se proporcionaron como entrada al modelo. Cambiar la forma de entrada del modelo de (40, 11) a

(20,11) y proporcionar el tensor resultante de un solo micrófono, permitió evaluar el rendimiento del modelo cuando se utiliza un solo micrófono. La tabla IV y la figura 5 proporcionan los resultados de validación cruzada de k veces cuando se utiliza un solo micrófono y demuestran claramente que el modelo funcionó mejor cuando se le da audio de dos micrófonos.

VII. DISCUSIÓN Y CONCLUSIÓN

Los hallazgos de este trabajo demuestran claramente que un CAE se puede utilizar con precisión para diferenciar los altavoces dado un pequeño tamaño de muestra de audio recogido de dos micrófonos simultáneamente. Con la validación cruzada, el modelo fue capaz de diferenciar entre dos, tres, cuatro, cinco y seis altavoces

TABLE IV: K-Fold Cross Validation Results – 1 Microphone

# of Buckets	Mean Accuracy (%)	Standard Deviation (%)
2	94.44	3.36
3	95.19	2.94
4	93.30	6.80
5	84.86	4.08
6	82.27	8.07

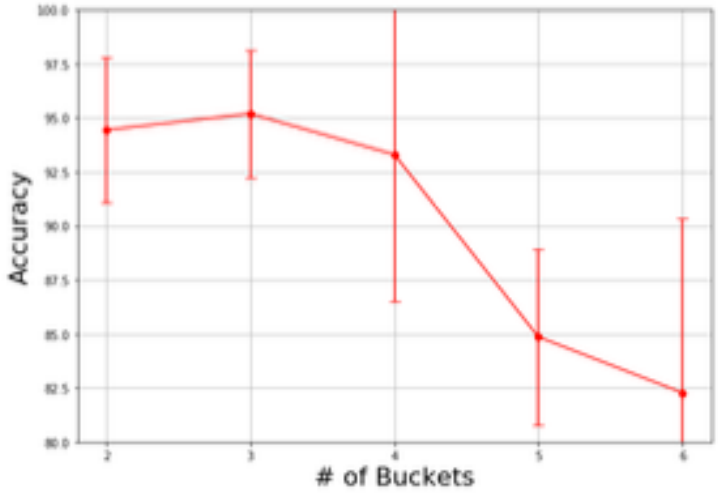

Fig. 5: KFold Cross Validation Results – 1 Microphone

Clases con una precisión de alrededor del 99 %, 97 %, 96 %, 93 % y 88 %, respectivamente. Los resultados también indican que a medida que aumenta el número de clases de oradores, la precisión del modelo disminuye. Esto confirma la hipótesis esbozada al comienzo de este trabajo de que no importa el tamaño y la calidad del conjunto de datos, a medida que aumenta

el número de cubos, la precisión del modelo eventualmente disminuiría. Sin embargo, dados los resultados de la precisión, cuando se les dieron hasta 5 altavoces diferentes, el modelo todavía fue capaz de diferenciar entre los altavoces en el audio recogido de dos micrófonos simultáneamente con una precisión de más del 90 %. Los resultados también demuestran que el modelo funcionó mejor cuando se le dio audio de dos micrófonos simultáneamente en comparación con un solo micrófono. Utilizando la validación cruzada y con datos de audio preprocesados recopilados de un solo micrófono, el modelo fue capaz de diferenciar entre dos, tres, cuatro, cinco y seis clases de altavoces con una precisión de aproximadamente el 94 %, el 95 %, el 93 %, el 85 % y el 82 %, respectivamente. Como tal, con el uso de dos micrófonos simultáneamente para recopilar el audio del altavoz, los resultados demuestran

que el uso de un CAE para resolver la clasificación de varios altavoces en el audio es tanto una solución eficiente como una solución de alto rendimiento.

Una investigación adicional sobre la caída en la precisión del modelo a medida que aumenta el número de cubos, podría incluir el uso de conjuntos de datos de tamaño de muestra más grande a medida que aumenta el número de cubos. Es posible que un aumento en el número de muestras de entrenamiento cuando se le da un mayor número de cubos pueda eliminar la caída en la precisión del modelo. Otros intentos de mejorar el rendimiento del modelo pueden incluir el aumento de la calidad del conjunto de datos de entrenamiento o el uso de otras técnicas de preprocesamiento de sonido que no sean MFCC. MFCC elimina una gran cantidad de información de la onda de audio original, utilizando otras técnicas que conServir más información, como la

transformación de Fourier a corto plazo (STFT, por sus siglas en inglés) que puede permitir que el modelo diferencie con mayor precisión entre los altavoces en el audio.

VIII. AGRADECIMIENTOS

Reconocemos el apoyo del Consejo de Investigación de Ciencias Naturales e Ingeniería de Canadá (NSERC) y de la Corporación Unificada de Inteligencia Informática de Canadá.

REFERENCIAS

[1] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe y J. R. Hershey, "Separación de múltiples altavoces de un solo canal mediante agrupación profunda", arXiv preprint arXiv:1607.02173, 2016.

[2] K. Lu, H.-W. Shen et al., "Análisis y visualización de datos volumétricos

multivariantes a través de la exploración del subespacio de abajo hacia arriba", en el Simposio de Visualización del Pacífico IEEE 2017 (PacificVis). IEEE, 2017, pp. 141-150.

[3] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, y P.-A. Manzagol, "Autocodificadores de desnuido apilados: Aprendizaje de representaciones útiles en una red profunda con un criterio de desnuido local", Journal of machine learning research, vol. 11, no. Dic, pp. 3371-3408, 2010.

[4] J. Masci, U. Meier, D. Cires¸an, y J. Schmidhuber, "Codificadores automáticos apilados para la extracción de características jerárquicas", en la Conferencia Internacional sobre Redes Neuronales Artificiales. Springer, 2011, pp. 52-59.

[5] B. Logan et al., "Mel frequency

cepstral coefficients for music model- ing." en ISMIR, vol. 270, 2000, pp. 1-11.

[6] L. Gondara, "Denoido de imagen médica utilizando autocodificadores de denoido convolucional", en Talleres de Minería de Datos (ICDMW), 16a Conferencia Internacional IEEE 2016 sobre. IEEE, 2016, pp. 241-246.

[7] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, y F. Chen, "Clasificación de imágenes sar de alta resolución a través de autocodificadores convolucionales profundos", IEEE Geoscience and Remote Sensing Letters, vol. 12, no. 11, pp. 2351-2355, 2015.

[8] P. Shree, P. Swami, V. Suresh y T. K. Gandhi, "Una nueva técnica para identificar la selección de atención en un entorno dicótico", en la Conferencia de la India (INDICON), IEEE Annual 2016. IEEE, 2016, pp. 1-5.

[9] M. Kolbæk, D. Yu, Z.-H. Tan y J. Jensen, "Separación conjunta y desnivel de ruido del discurso ruidoso de múltiples habladores utilizando redes neuronales recurrentes y entrenamiento de invariantes de permutación", en Machine Learning for Signal Processing (MLSP), 2017 IEEE 27th International Workshop on. IEEE, 2017, pp. 1-6.

[10] S. Albawi, T. A. Mohammed y S. Al-Zawi, "Comprensión de una red neuronal convolucional", en Ingeniería y Tecnología (ICET), Conferencia Internacional de 2017 sobre. IEEE, 2017, pp. 1-6.

[11] B. Yang, X. Fu y N. D. Sidiropoulos, "Aprendizaje de rasgos ocultos: análisis factorial conjunto y agrupación latente", IEEE Transactions on Signal Processing, vol. 65, no. 1, pp. 256-269, 2017.

[12] Y. Zheng, Q. Liu, E. Chen, Y. Ge, y J.

L. Zhao, "Clasificación de series temporales utilizando redes neuronales convolucionales profundas multicanal", en la Conferencia Internacional sobre Gestión de la Información de la Era Web. Springer, 2014, pp. 298-310.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever y R. Salakhut-dinov, "Dropout: una forma sencilla de evitar que las redes neuronales se ajusten en exceso", The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929-1958, 2014.

[14] D.-A. Clevert, T. Unterthiner y S. Hochreiter, "Aprendizaje profundo en red rápido y preciso por unidades lineales exponenciales (elus)", arXiv preprint arXiv:1511.07289, 2015.

[15] R. A. Dunne y N. A. Campbell, "Sobre el emparejamiento de las funciones de activación softmax y penalización de

entropía cruzada y la derivación de la función de activación softmax", en Proc. 8th Aust. Conf. sobre las redes neuronales, Melbourne, vol. 181. Citeseer, 1997, p. 185.

[16] S. Ruder, "Una visión general de los algoritmos de optimización de descenso de gradiente, 2016", arXiv preprint arXiv:1609.04747, 2016.

[17] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, y O. Nieto, "librosa: Análisis de señales de audio y música en python", en Proceedings of the 14th python in science conference, 2015, pp. 18-25.

[18] V. Sesiones y M. Valtorta, "Los efectos de la calidad de los datos en los algoritmos de aprendizaje automático". ICIQ, vol. 6, pp. 485-498, 2006.

[19] T. Wong y N. Yang, "Análisis de

dependencia de las estimaciones de precisión en la validación cruzada k-fold", IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 11, pp. 2417-2427, noviembre de 2017.