

Utilização de Aprendizado de Máquina na Demodulação de Sistemas OAM utilizando Multiplexação de Modos Orbitais

Bruno S Freitas, Cristhof J R Runge, Jaime Portugheis, Ivan Oliveira, Cecilio Pimentel e Ulisses Dias

Resumo— Este trabalho apresenta resultados de simulação para um sistema de transmissão óptica OAM no espaço livre sob efeito de turbulência atmosférica. Os momentos angulares orbitais, correspondendo aos símbolos de transmissão, são utilizados para modular um feixe de luz no modo *Laguerre-Gauss*. Na recepção, imagens simulando o que seria capturado por uma câmera CCD são analisadas por uma rede neural convolucional (CNN), o que corresponde ao processo de demodulação e classificação dos símbolos recebidos. Foram simuladas as transmissões de alfabetos utilizando superposição simples e utilizando tuplas formadas por multiplexação de superposições simples de dois momentos orbitais. Os resultados de simulação apontam desempenho similar dos dois casos nas condições de turbulência fraca e moderada. Este fato pode conduzir a possíveis ganhos de capacidade para sistemas OAM que utilizem alfabetos obtidos através da multiplexação de superposições simples de momentos angulares orbitais.

Palavras-Chave— Momento angular orbital, rede neural convolucional, aprendizado de máquina, turbulência atmosférica.

Abstract— This work presents the simulation results for an OAM optical transmission system in the free space under the effect of atmospheric turbulence. The orbital angular moments, corresponding to the transmission symbols, are used to modulate a light beam in Laguerre-Gauss mode. At the reception side, images simulating what is captured by a CCD camera are analyzed by a convolutional neural network in a process of demodulation and classification of received symbols. Simulations are performed using transmission of alphabets formed by tuples obtained by simple superposition of orbital modes, and by multiplexing two superposed orbital modes. The simulation results point to similar performance for the cases of systems using simple superposition and using tuples formed by multiplexing moments in the conditions of weak and moderate turbulence. These results may lead to possible capacity gains for OAM systems using alphabets obtained through multiplexing simple superposed orbital angular moments.

Keywords— Orbital angular momentum, convolutional neural network, machine learning, atmospheric turbulence.

I. INTRODUÇÃO

O aprendizado de máquina vem sendo amplamente utilizado na tarefa de reconhecimento de imagens. Mais especificamente, as redes neurais convolucionais CNN (do inglês *Convolutional Neural Network*) têm se mostrado uma solução eficiente para extração de informações a partir da análise de imagens, sendo utilizadas em diversas áreas do conhecimento com esta finalidade, tais como: área de saúde no diagnóstico

Bruno Freitas, Cristhof J R Runge, Ulisses Dias, Jaime Portugheis, and Ivan Oliveira are with the School of Technology, Unicamp, Limeira, Brazil; Cecilio Pimentel is with DES-UFPE, Recife, Brazil

por análise de imagens, em sistemas de sensoriamento remoto para identificação de regiões de interesse, ou ainda na área de visão computacional para reconhecimento de objetos, entre outras aplicações.

O emprego de aprendizado de máquina para demodulação adaptativa de um sistema OAM (do inglês *Orbital Angular Momentum*) foi recentemente proposto em [8]. Os autores apresentaram resultados de desempenho para a simulação de diferentes condições de turbulência atmosférica, diferentes distâncias e diferentes algoritmos de aprendizado de máquina. Em [3], foram obtidos resultados experimentais de laboratório através da construção de um sistema utilizando um modulador espacial SLM (do inglês *Spacial Light Modulator*) [9] para multiplexação de momentos orbitais em um feixe de luz no modo Bessel-Gauss. Uma câmera CCD (do inglês *Charge Couple Device*) foi utilizada na recepção para captura das imagens e posterior processamento por uma CNN. Os resultados experimentais mostram ganhos de desempenho em relação à abordagem tradicional de demultiplexação através da metodologia de ordenação por modo conjugado (do inglês *conjugate mode sorting*), além da simplificação do hardware de recepção.

Este trabalho compara o desempenho de diferentes sistemas de transmissão OAM utilizando aprendizado de máquina na recepção, quando os alfabetos de modulação são obtidos através da formação de diferentes símbolos a partir da multiplexação de modos orbitais $\pm l$ superpostos. O uso de modos superpostos se justifica devido a esta configuração apresentar maior robustez aos efeitos da turbulência atmosférica [11]. Com o intuito de avaliar o impacto de desempenho em diferentes condições de turbulência atmosférica, a imagem transmitida é corrompida através da adição de uma matriz de turbulência construída a partir da utilização do espectro de Kolmogorov. O desempenho dos sistemas para diferentes níveis de turbulência (fraca, moderada e forte) são verificados. Os resultados de simulação mostram que, para situações de turbulência fraca e moderada, o desempenho dos sistemas utilizando símbolos multiplexados são similares aos do sistema utilizando superposição simples, o que aponta para um ganho de capacidade, uma vez que a CNN se mostra capaz de distinguir entre símbolos em um alfabeto de maior cardinalidade (possível devido ao maior número de combinações para um mesmo conjunto de modos orbitais).

O restante deste texto está organizado da seguinte forma: a Seção II apresenta a teoria utilizada para a geração de modos OAM em um feixe Laguerre-Gauss, a Seção III discute

a implementação da matriz de turbulência utilizando o espectro de Kolmogorov, a Seção IV apresenta o modelo de CNN utilizado, a Seção V apresenta a metodologia utilizada nas simulações para os diferentes sistemas, cenários e os resultados obtidos. A Seção VI finaliza o artigo.

II. CONCEITOS TEÓRICOS PARA GERAÇÃO DE MOMENTOS ANGULARES ORBITAIS

Os momentos angulares orbitais podem ser produzidos através de manipulação do perfil elétrico transversal de um feixe de luz. Por sua vez, o campo elétrico de uma onda eletromagnética é solução da equação de Helmholtz:

$$\nabla^2 E + k^2 E = 0 \quad (1)$$

onde ∇^2 é o operador Laplaciano, E representa o campo elétrico e $k = 2\pi n\lambda_0$, sendo n o índice de refração no meio de propagação e λ_0 o comprimento de onda no espaço livre.

O campo elétrico E pode ser escrito de uma forma mais conveniente como uma função do tempo e o correspondente versor:

$$\vec{E}(\vec{r}, t) = \vec{e}u(\vec{r}) \exp(-j\omega t) \quad (2)$$

onde \vec{e} representa o versor na direção do campo elétrico, u é uma função escalar correspondendo à intensidade de campo em um ponto do espaço definido pelo vetor posição \vec{r} e o termo exponencial corresponde à variação temporal do campo elétrico. Para a análise espacial do feixe, a componente temporal pode ser suprimida (similarmente ao que ocorre em análise fatorial) e a Equação 2 pode ser expressa apenas em termos das componentes espaciais como:

$$\vec{E}(\vec{r}) = \vec{e}\psi(x, y, z) \exp(-jkz) \quad (3)$$

Nesta equação, a função escalar que representa o perfil de intensidade espacial do campo elétrico $u(\vec{r})$ é desmembrada na função $\psi(x, y, z)$ e na contribuição da variação espacial $\exp(-jkz)$ ao longo do eixo de propagação z do feixe de interesse. Aplicando a equação de Helmholtz à função $u(\vec{r})$ temos:

$$\nabla^2 u(\vec{r}) + k^2 u(\vec{r}) = 0 \quad (4)$$

que pode ser escrita como:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + k^2 u = 0 \quad (5)$$

A. Aproximação Paraxial

Considerando que a dependência em relação a z no plano transversal é muito pequena devido à pequena área da seção transversal e ao raio de curvatura do feixe, podemos assumir que:

$$\left| \frac{\partial^2 \psi(x, y, z)}{\partial z^2} \right| \ll \left| \frac{2k \partial \psi(x, y, z)}{\partial z} \right| \quad (6)$$

Com estas considerações, assumindo que a variação ao longo do eixo de propagação é muito pequena em relação

ao raio de curvatura transversal do feixe, chega-se na equação paraxial para a intensidade do campo elétrico da onda:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} - 2jk \frac{\partial \psi}{\partial z} = 0 \quad (7)$$

Existem diversas soluções para a Equação 7, como os modos Gaussiano, Hermite-Gauss e Laguerre-Gauss. Este último representado abaixo através da equação 8.

$$\psi_p^l(x, y, z) = \frac{A_p^l}{\omega(z)} \left[\frac{\sqrt{2(x^2 + y^2)}}{\omega(z)} \right]^{|l|} L_p^l \left(\frac{2(x^2 + y^2)}{\omega^2(z)} \right) \exp \left(\frac{x^2 + y^2}{\omega^2(z)} \right) \times \exp \left\{ -i \left[\frac{k(x^2 + y^2)}{2R(z)} \right] + (2p + |l| + 1) \tan^{-1} \left(\frac{z}{z_R} \right) + l \tan^{-1} \left(\frac{y}{x} \right) \right\} \quad (8)$$

onde A_p^l é uma constante de normalização, L_p^l é o polinômio generalizado de Laguerre, l é a ordem do modo orbital OAM, $\omega(z) = w_0 \sqrt{1 + (z/z_R)^2}$ é o raio do feixe na distância z , w_0 é a cintura do feixe, $R(z) = z \left[1 + \left(\frac{z_R}{z} \right)^2 \right]$ é o raio de curvatura na distância z , $z_R = \pi \frac{w_0^2}{\lambda}$ é a distância de Rayleigh e λ é o comprimento de onda.

Para os sistemas simulados neste artigo, foram usados os modos Laguerre-Gauss com $p = 0$ e com $l \in \mathbb{Z}$. Note que para $p = 0$ o polinômio generalizado de Laguerre se reduz a $L_0^l = 1$. Como os símbolos dos alfabetos de modulação serão formados por superposições simples $\pm l$ de modos orbitais, ou multiplexação de superposições simples, um símbolo de modulação pode ser descrito de forma geral como:

$$S_L(x, y, z) = \sum_{l \in L} (\psi_0^{+l} + \psi_0^{-l}) \quad (9)$$

onde L representa o subconjunto de modos escolhidos para formação de um dado símbolo de transmissão, escolhidos dentro do conjunto de todos os possíveis modos orbitais disponíveis para a formação de um determinado alfabeto.

Em um sistema real, um símbolo de transmissão conforme o descrito pela Equação 9 pode ser obtido através da passagem de um feixe de luz em modo fundamental (gaussiano) por uma rede de difração ou através do uso de um SLM. Um SLM é um dispositivo que possui uma tela de cristal líquido capaz de modular um feixe gaussiano incidente através de um processo de difração. A partir da reprodução de um holograma, correspondendo a uma máscara de fase na superfície da tela, o feixe incidente é difratado (modulado) e passa a carregar os modos orbitais correspondentes.

III. TURBULENCIA ATMOSFÉRICA

O efeito de turbulência é causado devido a propagação do feixe pela atmosfera. As flutuações no índice de refração atmosférico durante a propagação produzem dispersões no feixe óptico provocando degradação do sinal recebido. Em sistemas OAM, o sinal a ser demodulado consiste em uma imagem do perfil de intensidade do feixe recebido e a turbulência atmosférica causará distorções na imagem recebida. Os modelos espectrais que tem sido mais amplamente utilizados para reproduzir os efeitos da turbulência atmosférica são baseados em aproximações do espectro de Kolmogorov [7].

$$\Phi(k_x, k_y) = 0.023(2D/r_0)^{\frac{5}{3}} (k_x^2 + k_y^2)^{-\frac{11}{3}} \quad (10)$$

onde $\Phi(k_x, k_y)$ é a função de aproximação da densidade espectral de potência de Kolmogorov, r_0 é o parâmetro de Fried e D é o comprimento em uma dimensão de uma abertura bidimensional quadrada da seção transversal do feixe recebido.

De forma a reproduzir os efeitos da turbulência atmosférica para fins de simulação, a abordagem normalmente utilizada é a criação de uma matriz de turbulência, também denominada *random phase screen*, que insere uma componente de fase aleatória em cada *pixel* da imagem recebida. Esta matriz é formada a partir da transformada inversa bidimensional de Fourier da função de aproximação do espectro de Kolmogorov pré-multiplicada por uma matriz de ruído conforme abaixo.

$$\phi(x, y) = IFFT \left(C \sqrt{\Phi(k_x, k_y)} \right) \quad (11)$$

onde $IFFT$ é a transformada inversa bidimensional de Fourier e C representa a matriz de ruído formada por variáveis aleatórias gaussianas complexas com o mesmo número de elementos correspondente ao número de *pixels* da imagem do símbolo recebido. A Figura 1 é uma representação de uma matriz de turbulência.

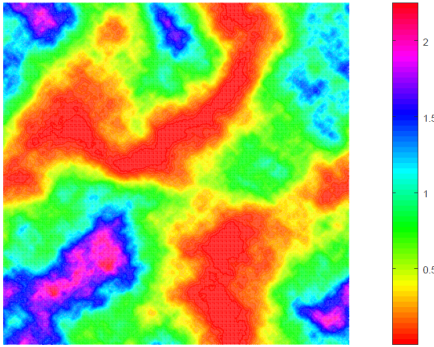


Fig. 1. Exemplo de uma realização da matriz de turbulência utilizando a equação 11; escala em radianos

O sinal na recepção após a inserção da matriz de turbulência pode então ser escrito como:

$$R(x, y) = \exp(i\phi(x, y)) \sum_{l \in L} (\psi_0^{+l} + \psi_0^{-l}) \quad (12)$$

sendo que a imagem do perfil de intensidade correspondente será dada por:

$$I_R(x, y) = |R(x, y)|^2 \quad (13)$$

A partir dos valores da relação D/r_0 , é possível variar os níveis de turbulência. Neste artigo, são investigados os resultados de simulação para sistemas submetidos a três tipos de turbulência: fraca, moderada e forte. A turbulência fraca foi simulada atribuindo-se valor $D/r_0 = 0.0005$ na Equação 10, a turbulência moderada usa $D/r_0 = 0.003$ e a turbulência forte usa $D/r_0 = 0.009$.

IV. REDES NEURAIS CONVOLUCIONAIS

Rede neural convolucional (CNN) é uma modificação da rede neural de multicamadas de neurônios, pois substitui a

multiplicação de matrizes pela operação de convolução [5]. Convoluções são operações matemáticas capazes de evidenciar padrões em imagens. Elas agem como filtros que capturam aspectos específicos. Matematicamente, a convolução é uma operação definida pela Equação 14. No âmbito de CNN, a convolução recebe duas matrizes como entrada e produz uma terceira matriz como saída.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (14)$$

Imagens são matrizes com três dimensões: altura, largura e canais de cores. Para realizar convoluções, são necessárias duas matrizes, sendo a primeira delas um canal da imagem de entrada e a segunda uma matriz chamada de *kernel*. O resultado da convolução é uma nova matriz conhecida como mapa de características. A Figura 2 esquematiza a convolução entre uma imagem 4×4 e um *kernel* 2×2 , o que resulta em um mapa de características 3×3 .

Imagem de Entrada				Kernel	Mapa de Características						
a	b	c	d	<table border="1"> <tr><td>x</td><td>y</td></tr> <tr><td>z</td><td>w</td></tr> </table>	x	y	z	w	$ax+by+ez+fw$	$bx+cy+fz+gw$	$cx+dy+gz+hw$
x	y										
z	w										
e	f	g	h		$ex+fy+ix+jw$	$fx+gy+jz+kw$	$gx+hy+kz+lw$				
i	j	k	l	$ix+iy+mz+nw$	$jx+ky+nz+ow$	$kx+ly+oz+pw$					
m	n	o	p								

Fig. 2. Representação Visual de uma Convolução. Uma imagem de entrada 4×4 é processada por um *kernel* 2×2 , que funciona como filtro para gerar um mapa de características 3×3 .

A arquitetura CNN é composta por camadas convolucionais contendo múltiplos *kernels* aplicados na imagem de entrada (ou no resultado da camada anterior) para a extração de características [4]. Alguns *kernels* são capazes de realçar contornos, outros suavizam a imagem, outras detectam bordas verticais enquanto que outras detectam as horizontais e assim por diante.

Os parâmetros que compõem um *kernel* não estão fixos, eles são “aprendidos” na fase de treinamento de acordo com a tarefa que se pretende resolver. Esse treinamento simplifica a criação do modelo, pois não é necessário um conhecimento tão aprofundado do problema a ponto de saber *a priori* quais os *kernels* necessários.

Os resultados da operação de convolução em cada camada passam por uma transformação não linear chamada de função de ativação. Essa etapa é considerada como fase de detecção [5]. Algumas escolhas comuns para a função de ativação são a tangente hiperbólica (função não linear suave no intervalo entre -1 e 1), a função sigmóide (função não linear suave no intervalo 0 e 1) e a função ReLU (função não linear que transforma em zero todos os elementos negativos). A função de ativação aumenta o poder de representação de uma rede neural.

Neste trabalho, foi utilizada a função ReLU em todas as camadas intermediárias (Equação 15) e a função softmax na camada de saída (Equação 16). Note que na Equação 16 existem K nós na camada de saída e o valor devolvido para cada um dos nós depende dos outros na mesma camada. Isso permite fazer com que apenas um desses nós seja ativado em um dado momento, sendo este nó o representante da classe que a rede está prevendo.

$$\text{ReLU}(x) = \max(0, x) \quad (15)$$

$$\text{Softmax}(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ para } j = 1 \dots K \quad (16)$$

As operações de *pooling* são aplicadas nos vários mapas de características e combinam células em uma única saída, resultando na diminuição do tamanho das camadas. Isso tende a diminuir a complexidade do modelo e previne que o modelo memorize os dados de treinamento (*overfitting*), o que resultaria em um modelo incapaz de generalizar para novas entradas. O *pooling* mais comum é chamado de *max-pooling*, que define um bloco (por exemplo, 3×3) e seleciona o maior elemento do mapa de características após a aplicação da função de ativação neste bloco. A Figura 3 ilustra a aplicação de *max-pooling* em um mapa de característica.

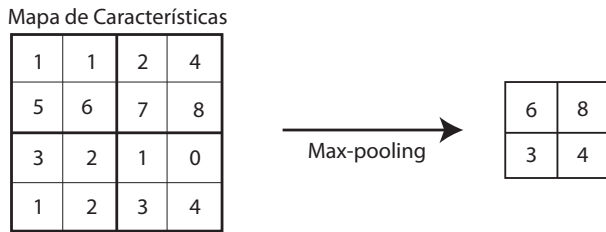


Fig. 3. Aplicação de *max-pooling* reduzindo a dimensionalidade de um mapa de características.

A aplicação de diversas camadas de *pooling* e convolução permite ao final gerar uma série de números em ponto flutuante que representam as principais informações da imagem de entrada. A partir dessas informações, é possível a modelagem de problemas específicos. No caso deste trabalho, o objetivo é verificar quais modos orbitais foram multiplexados para gerar a imagem, sabendo que o sinal pode ter sido degradado por meio de turbulência. Em outras palavras, o objetivo é classificar a imagem em um conjunto de rótulos que compõem o alfabeto utilizado na transmissão da informação.

Neste trabalho, essa tarefa de classificação ficou a cargo de uma rede *Multilayer Perceptron* acoplada na saída das camadas de convolução. A rede completa apresentada neste trabalho pode ser dividida em duas partes: a primeira contendo convoluções e *poolings* para extração de características e a segunda contendo nós completamente conectados para demodulação da informação contida na imagem. A Figura 4 apresenta as camadas.

A entrada da rede consiste em imagens com dimensões 92×92 . A primeira camada possui 16 filtros de dimensões 5×5 , a segunda possui 32 filtros de dimensões 3×3 , a terceira possui 64 filtros de dimensões 3×3 e a quarta e última camada possui 128 filtros também de dimensões 3×3 . A função de ativação usada em todos os layers é a ReLU. Observe que entre uma camada e outra foi utilizado *Batch Normalization*, uma transformação que normaliza a saída da camada de modo a manter a média dos valores próxima de zero e o desvio padrão próximo de 1.

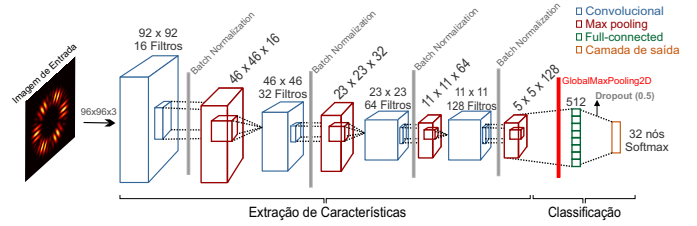


Fig. 4. Arquitetura CNN utilizada neste trabalho. Quatro camadas de convolução foram utilizadas para extração de características e uma camada completamente conectada foi utilizada para resolver o problema de classificação.

Ao final das camadas de extração de características, os valores são agrupados em um *array* unidimensional e fornecidos nesse formato a um rede neural artificial completamente conectada. Essa rede neural possui uma única camada interna com 512 nós e função de ativação ReLU. A camada de saída possui 32 nós e função de ativação *softmax*. Na camada de saída, apenas um nó ficará ativo a cada execução da rede e esse nó representa um elemento do alfabeto. Nesse caso, a ativação do nó é o resultado do processo de demodulação da imagem de entrada.

A implementação do modelo descrito foi feita na linguagem Python, com a biblioteca Keras [2] e o Tensorflow [1] como *backend*. O algoritmo *Adam* foi usado para a otimização do modelo com $\eta=0.001$.

V. METODOLOGIA E RESULTADOS

O banco de dados para obtenção dos resultados experimentais foi criado simulando três níveis de turbulência no sistema de comunicação: leve, moderada e forte. As imagens que simulam o que é recebido por uma câmera CCD na recepção sempre possuem um desses tipos de turbulências.

Quatro cenários foram simulados com alfabetos de 32 símbolos. Esses cenários diferem pelo modo como os modos OAM são multiplexados. No **Cenário 1**, os símbolos do alfabeto são compostos pela simples superposição de modos OAM $\pm l$ tal que $l \in \{1, 2, \dots, 31, 32\}$.

No **Cenário 2**, os símbolos usam multiplexação de dois modos OAM $\pm l_1$ e $\pm l_2$. Três sub-cenários foram criados tentando elucidar quais seriam as melhores escolhas para compor um alfabeto de 32 símbolos. No **Cenário 2.1**, foram selecionados valores baixos de $\pm l_1$ e $\pm l_2$: $\{(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (3, 4), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (4, 5), (4, 6), (4, 7), (4, 8), (4, 9), (5, 6), (5, 7), (5, 8), (5, 9), (6, 7), (6, 8)\}$.

No **Cenário 2.2**, foram escolhidos valores altos de $\pm l_1$ e $\pm l_2$: $\{(24, 29), (24, 30), (24, 31), (24, 32), (25, 26), (25, 27), (25, 28), (25, 29), (25, 30), (25, 31), (25, 32), (26, 27), (26, 28), (26, 29), (26, 30), (26, 31), (26, 32), (27, 28), (27, 29), (27, 30), (27, 31), (27, 32), (28, 29), (28, 30), (28, 31), (28, 32), (29, 30), (29, 31), (29, 32), (30, 31), (30, 32), (31, 32)\}$.

No **Cenário 2.3**, foram combinados valores altos e baixos: $\{(1, 32), (1, 31), (1, 30), (1, 29), (1, 28), (1, 27), (2, 32), (2, 31), (2, 30), (2, 29), (2, 28), (2, 27), (3, 32), (3, 31),$

(3, 30), (3, 29), (3, 28), (3, 27), (4, 32), (4, 31), (4, 30), (4, 29), (4, 28), (4, 27), (5, 32), (5, 31), (5, 30), (5, 29), (5, 28), (5, 27), (6, 32), (6, 31)}.

Para cada elemento nos alfabetos, 400 imagens foram geradas com turbulência leve, 400 imagens com turbulência moderada e 400 imagens com turbulência forte. Dessa forma, em cada cenário, 38400 imagens foram geradas e separadas em conjunto de treino e conjunto teste na proporção de 80% para treino e 20% para teste. A acurácia foi a métrica utilizada para verificar quão bem os modelos demodulam a imagem de entrada. Os resultados são apresentados na Tabela I.

TABELA I
ACURÁCIA DO MODELO DE REDE NEURAL CONVOLUCIONAL EM CADA UM DOS CENÁRIOS.

Cenários	Turbulência		
	Leve	Moderada	Forte
1	1.00	1.00	0.82
2.1	1.00	0.99	0.60
2.2	1.00	0.99	0.73
2.3	1.00	0.84	0.33

O Cenário 1 se apresentou mais resistente a variações de turbulência, obtendo 100% de acurácia com turbulência leve e moderada. Em situações de turbulência forte, o Cenário 1 foi superior aos demais com uma acurácia de 82%.

Os cenários 2.1 e 2.2 obtiveram excelentes resultados de acurácia com turbulências leve e moderada, mas um decréscimo ocorreu na turbulência forte. O Cenário 2.3 obteve resultados inferiores aos demais, com 84% de acurácia em turbulência moderada e 33% em turbulência forte.

Um estudo mais aprofundado ainda precisa ser feito para entender o motivo das variações de acurácia. É um fato que o Cenário 1 gera imagens mais simples que os alfabetos nos outros cenários. Nesse caso, um motivo para o melhor desempenho pode ser que o modelo de CNN utilizado é simples demais e o poder de representação não foi capaz de realizar a demodulação no Cenário 2. Um estudo com redes mais profundas pode elucidar essa hipótese.

VI. CONCLUSÕES

Este trabalho comparou o desempenho de um sistema usando um alfabeto OAM contendo a superposição simples de modos orbitais com três alfabetos distintos formados pela multiplexação par-a-par de superposições simples de modos orbitais. Todos os alfabetos testados possuem a mesma quantidade de elementos, possuindo mesma taxa (bits/símbolo) para a transmissão.

Os resultados para os cenários 2.1 e 2.2 se mostraram com desempenho praticamente igual ao cenário 1 para as condições de turbulência fraca e moderada, ou seja, a CNN foi capaz de distinguir entre símbolos formados por modos multiplexados de forma tão eficiente como de símbolos formados por superposição simples.

Este fato levanta um aspecto importante em termos de ganhos de capacidade para transmissão, uma vez que o número de combinações par-a-par possíveis para um conjunto OAM é muito superior ao tamanho do próprio conjunto.

No caso da superposição simples é possível mapear $\log_2 32 = 5$ bits por símbolo. No entanto, para o caso das multiplexações par-a-par, é possível criar alfabetos com até $\binom{32}{2} = 496$ símbolos, o que permite $\log_2 496 \cong 8.95$ bits por símbolo.

Como próximos passos, pretendemos aumentar a cardinalidade dos alfabetos formados por modos multiplexados. Isso pode gerar um ganho em termos de taxa de transmissão da informação caso a acurácia da CNN não diminua significativamente.

Neste trabalho, o problema foi tratado utilizando uma CNN de poucas camadas e aplicando a mesma rede em todos os cenários. Em trabalhos futuros, é preciso avaliar se as imagens geradas pelos diversos alfabetos possuem complexidades discrepantes a ponto de exigirem uma rede específica para cada cenário.

A CNN utilizada neste trabalho foi inicializada com valores aleatórios em cada cenário e o processo de treinamento foi realizado até o ponto em que a rede convergiu nos dados de treino. O treinamento de uma rede inicializada com valores aleatórios é um processo oneroso computacionalmente, o que explica o fato de não ter sido utilizada uma rede muito mais profunda.

Uma outra linha de pesquisa consiste em utilizar arquiteturas de redes neurais profundas já disseminadas na literatura, como DenseNet [6] ou VGG [10], para verificação dos resultados de desempenho. Essas redes possuem uma gama muito grande de variáveis para se atribuir peso durante o treinamento, o que impacta no tempo de processamento. O emprego da técnica de transferência de conhecimento pode aliviar ambos os problemas. Essa técnica consiste em reaproveitar em um segundo problema a seção de extração de características das redes treinadas em um problema qualquer. As principais redes já foram utilizadas em problemas diversos e com o resultado do treinamento amplamente disponível.

REFERÊNCIAS

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] François Chollet et al. Keras. <https://keras.io>, 2015.
- [3] Timothy Doster and Abbie T. Watnik. Machine learning approach to oam beam demultiplexing via convolutional neural networks. *Applied Optics*, 56(12):3386–3396, Apr 2017.
- [4] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* "O'Reilly Media, Inc.", 2017.
- [5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. <http://www.deeplearningbook.org>.
- [6] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.
- [7] RG Lane, A Glindemann, and JC Dainty. Simulation of a kolmogorov phase screen. *Waves in random media*, 2(3):209–224, 1992.
- [8] J. Li, M. Zhang, and D. Wang. Adaptive demodulator using machine learning for orbital angular momentum shift keying. *IEEE Photonics Technology Letters*, 29(17):1455–1458, Sep. 2017.
- [9] John A Neff, Ravinda A Athale, and Sing H Lee. Two-dimensional spatial light modulators: a tutorial. *Proceedings of the IEEE*, 78(5):826–855, 1990.
- [10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [11] Yangsheng Yuan, Ting Lei, Gao, et al. The orbital angular momentum spreading for cylindrical vector beams in turbulent atmosphere. *IEEE Photonics Journal*, 9(2):1–10, 2017.